Summer 2016

Machine Learning Methods for Brain Image Analysis

Ahmed Fakhry
Old Dominion University

MACHINE LEARNING METHODS FOR BRAIN IMAGE ANALYSIS

by

Ahmed Fakhry
B.S. June 2009, Alexandria University, Egypt

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2016

Approved by:

Shuiwang Ji (Director)

Andrey Chernikov (Member)

Jing He (Member)

Dean Krusienski (Member)

ABSTRACT

MACHINE LEARNING METHODS FOR BRAIN IMAGE ANALYSIS

Ahmed Fakhry

Old Dominion University, 2016

Director: Dr. Shuiwang Ji

Understanding how the brain functions and quantifying compound interactions between complex synaptic networks inside the brain remain some of the most challenging problems in neuroscience. Lack or abundance of data, shortage of manpower along with heterogeneity of data following from various species all served as an added complexity to the already perplexing problem. The ability to process vast amount of brain data need to be performed automatically, yet with an accuracy close to manual human-level performance. These automated methods essentially need to generalize well to be able to accommodate data from different species. Also, novel approaches and techniques are becoming a necessity to reveal the correlations between different data modalities in the brain at the global level. In this dissertation, I mainly focus on two problems: automatic segmentation of brain electron microscopy (EM) images and stacks, and integrative analysis of the gene expression and synaptic connectivity in the brain. I propose to use deep learning algorithms for the 2D segmentation of EM images. I designed an automated pipeline with novel insights that was able to achieve state-of-the-art performance on the segmentation of the *Drosophila* brain. I also propose a novel technique for 3D segmentation of EM image stacks that can be trained end-to-end with no prior knowledge of the data. This technique was evaluated in an ongoing online challenge for 3D segmentation of neurites where it

achieved accuracy close to a second human observer. Later, I employed ensemble learning methods to perform the first systematic integrative analysis of the genome and connectome in the mouse brain at both the regional- and voxel-level. I show that the connectivity signals can be predicted from the gene expression signatures with an extremely high accuracy. Furthermore, I show that only a certain fraction of genes are responsible for this predictive aspect. Rich functional and cellular analysis of these genes are detailed to validate these findings.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Shuiwang Ji, for his tremendous efforts and support throughout my years of research. I am indebted to him for the confidence and passion he demonstrated in my work. Working with him is an experience that I will always cherish and be grateful for. I would like to thank my defense committee members, Dr. Andrey Chernikov, Dr. Jing He and Dr. Dean Krusienski, for their insightful feedback and interactions.

I have been lucky to work on several large-scale projects which involved a group of highly motivated students from which I have gained a lot technically and personally. I would like to thank my collaborators, Rongjian Li, Tao Zeng and Wenlu Zhang for their insightful discussions and valuable interactions.

Last but not least, my family, this work would not have been done without them. I want to thank my wife Maryam for the unconditional support she continuously provides, my son, my mother and sisters. Thanks to my father for setting the example that I aspire to become.

My research work is supported in part by the National Science Foundation grant DBI-1350258 and ODU Office of Research.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Advancements in neuroscience is strongly impacted by the continuous progression in the computer science field. Over the past decades, data availability was the main factor in our limited understanding of neurological systems and its complicated structures across various species. This limitation has been addressed recently by the rapid improvements in data acquisition techniques which lead to an abundance of data ready to be mined. However, mining such vast amounts of data is overwhelmingly labor intensive. In addition, relying solely on manpower in such tasks is highly and understandably error prone. This highlights the apparent urge for automated machine-based techniques to help reduce the gap between how much data is available to us and the amount of information we can extract from it. This also raises the questions of how much can powerful machine-based data processing and mining techniques contribute to the neuroscience field? Can it eventually provide some answers and insights to one of the most fundamentally challenging questions of how the brain works?

In this thesis, I approach this relationship between computer science and neuroscience. I aspire to bridge aforementioned gap by providing automated machine learning based methodologies to some of the critically important problems in neuroscience. In the techniques I propose in this thesis, I rely on computer science and specifically machine learning algorithms with a direct biological application to neuroscience domains. The goal is to push the progression of computational methods side by side with the amount of data available towards a better understanding of our neural system.

Specifically, this thesis is divided into two sections answering two different questions: First, can we improve upon the existing state-of-the-art data acquisition and processing techniques to provide the data in a format that is ready to mine? Second, assuming that the data is available, what kind of information and insights can we retrieve from it? In the next few sections, I will provide a brief description of my attempts to answer each of those questions.

1.1 AUTOMATIC SEGMENTATION OF ELECTRON MICROSCOPY IMAGES

In this section of the thesis, I used deep learning in an attempt to solve the problem of accurate segmentation of brain electron microscopy (EM) images, which is a critical step in dense circuit reconstruction. Typically, reconstructing brain circuits begins by identifying individual neurons first by segmenting them accurately. Later, axons and dendrites are reconstructed to elucidate inter-neuron connections. This segmentation process is very challenging, time- and labor-intensive. For example, it took 15 years for researchers to fully reconstruct the nervous system of the worm *Caenorhabditis elegans* manually. Performing the same process manually on more complex species like mouse or human is clearly infeasible. Consequently, automated methods for image segmentation with an accuracy that approaches or preferably exceeds human accuracy are inevitably needed.

I start by providing a brief introduction about deep learning and its applications in the computer vision field in Chapter 2. Chapter 3 then provides my first attempt to use deep neural networks (DNNs) for 2D EM images segmentation. While DNNs have been widely used in a number of applications in computer vision, most of these models that proved to be effective on image classification tasks cannot be applied

directly to EM image segmentation, due to the different objectives of these tasks. As a result, it is desirable to develop an optimized architecture that uses the full power of DNNs and tailored specifically for EM image segmentation. I proposed a novel design of DNNs for this task. I trained a pixel classifier that operates on raw pixel intensities with no preprocessing to generate probability values for each pixel being a membrane or not. While the use of neural networks in image segmentation is not completely new, I developed novel insights and model architectures that allowed me to achieve superior performance on EM image segmentation tasks. My submission based on these insights to an online 2D EM Image Segmentation Challenge achieved the best performance consistently across all three evaluation metrics. The accuracy of my proposed model was one of few that exceeded the accuracy of a second human observer. This challenge is still ongoing and the results in this thesis are as of June 5th, 2015.

In Chapter 4, I extend my experiments to the 3D domain by attempting to segment 3D EM image stacks. Despite the recent technical advances in EM image segmentation, most of them rely on hand-crafted features to some extent that are unique to the data, limiting their ability to generalize. In this chapter, I propose a simple yet powerful technique for EM image segmentation that is trained end-to-end and does not rely on any prior knowledge of the data. My proposed residual deconvolutional network consists of two information pathways that capture full-resolution features and contextual information, respectively. I show that the proposed model is very effective in achieving the conflicting goals in dense output prediction of preserving a full-resolution prediction with the inclusion of sufficient contextual information. I applied my method to the ongoing open challenge of 3D neurites segmentation in electron microscopy images, and my method achieved very promising results. I demonstrate the generality of my technique by evaluating on the ISBI 2012 challenge

dataset where I obtained consistently high performance. I thus expect my method to generalize well to other dense output prediction problems.

1.2 INTEGRATIVE ANALYSIS OF GENOME AND CONNECTOME IN THE MOUSE BRAIN

Brain function is the result of interneuron signal transmission controlled by the fundamental biochemistry of each neuron. The biochemical content of a neuron is in turn determined by spatiotemporal gene expression and regulation encoded into the genomic regulatory networks. It is thus of particular interests to elucidate the relationship between gene expression patterns and connectivity in the brain. However, systematic studies of this relationship in a single mammalian brain are lacking to date. After demonstrating the ability to improve upon the existing methods to obtain brain connectivity data in Chapters 3 and 4, I proceed to provide examples of applications that can be carried on using this data modality in this section of my thesis.

In Chapter 5, I investigate the relationship between the genome and connectome in the mouse brain using the Allen Brain Atlas data. I employ computational models for predicting brain connectivity from gene expression data. In addition to giving competitive predictive performance, these models can rank the genes according to their predictive power. I show that gene expression is predictive of connectivity in the mouse brain when the connectivity signals are discretized. When the expression patterns of 4084 genes are used, I obtain a predictive accuracy of 93%. My results also show that a small number of genes can almost give the full predictive power of using thousands of genes. I can achieve a prediction accuracy of 91% by using only 25 genes. Gene ontology analysis of the highly ranked genes shows that they are enriched for connectivity related processes.

Chapter 6 investigates the same relationship between genome and connectome but

in a high-resolution analysis. Currently, studies in multiple species have indicated that the spatiotemporal gene expression patterns are predictive of brain wiring. Specifically, results on the worm *Caenorhabditis elegans* have shown that the prediction of neuronal connectivity using gene expression signatures yielded statistically significant results. Recent studies on the mammalian brain produced similar results at the coarse regional level. In this chapter, I provide the first high-resolution, large-scale integrative analysis of the transcriptome and connectome in a single mammalian brain at a fine voxel level. By using the Allen Brain Atlas data, I predict voxel-level brain connectivity based on the gene expressions in the adult mouse brain. I employ regularized models to show that gene expression is predictive of connectivity at the voxel-level with an accuracy of 93%. I also identify a set of genes playing the most important role in connectivity prediction. I use only this small number of genes to predict the brain wiring with an accuracy over 80%. I discover that these important genes are enriched in neurons as compared to glia, and they perform connectivity-related functions. I perform several interesting correlative studies to further elucidate the transcriptome-connectome relationship.

Chapter 7 provides a global, integrative analysis of gene expression and projection target correlations in the mouse brain. Recent studies have shown that projection targets in the mouse neocortex are correlated with their gene expression patterns. However, a brain-wide quantitative analysis of the relationship between voxel genetic composition and their projection targets is lacking to date. In this chapter, I used the Allen Brain Atlas data to analyze the relationship between gene expression and projection targets. I first visualized and clustered the two data sets separately and showed that they both exhibit strong spatial autocorrelation. Building upon this initial analysis, I conducted an integrative correlation analysis of the two data sets while correcting for their spatial autocorrelation. This resulted in a correlation of

0.19 with significant p-value. I further identified the top genes responsible for this correlation using two greedy gene ranking techniques. Using only the top genes identified by those techniques, I recomputed the correlation between these two data sets. This led to correlation values up to 0.49 with significant p-values. My results illustrated that although the target specificity of neurons is in fact complex and diverse, yet they are strongly affected by their genetic and molecular compositions. I conclude my thesis in Chapter 8 by providing a summary of the contributions and directions for future work.

CHAPTER 2

DEEP LEARNING IN COMPUTER VISION

In this chapter, I provide a brief description of the techniques I employed in Chapters 3 and 4. I list the key terminology used along with several applications in computer vision. I then describe convolutional neural networks which acts as a building block for most deep learning applications. I describe the most common layers used, popular architectures along with some of the best performances achieved on some datasets.

2.1 DEEP LEARNING

Deep learning is defined as a family of machine learning techniques which attempt to learn different high-level data abstractions using a series of non-linear transformations. These algorithms generally rely on complex model architectures that can achieve a better representations from large-scale unlabeled data. An algorithm is considered deep if the input passes through several layers (non-linearities) before producing an output. Most of the current popular machine learning techniques like SVM and logistic regression are shallow.

The power of deep learning is illustrated in its deep architectures which can more gracefully reuse previous computations. More complex features are extracted from the input while passing through several layers of assessment. The goal is to replace handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction. As a result, these models have a great potential to generalize to different tasks, moving us one step closer to Artificial Intelligence.

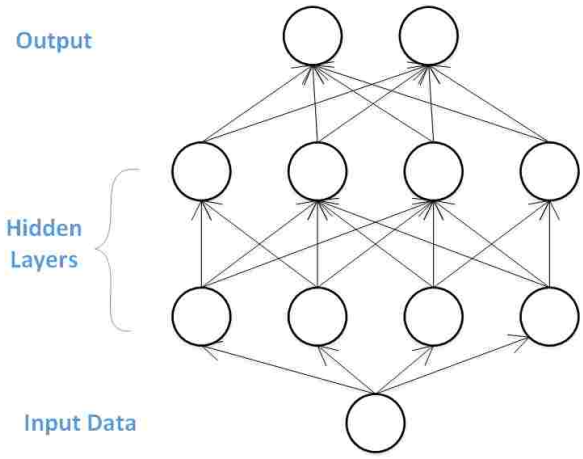


FIG. 1: A typical CNN with 2 hidden layers.

Deep learning techniques have recently dominated the computer vision field with applications spanning a wide range of topics including image classification, object and scene recognition, handwriting detection and image segmentation. It has also shown state-of-the-art performances on other tasks like speech recognition, natural language processing and bioinformatics. The most popular of those techniques are convolutional neural networks, deep belief networks and recurrent neural networks.

2.2 CONVOLUTIONAL NEURAL NETWORKS

The convolution neural networks (CNN) are the most popular deep learning technique with tremendous applications in computer vision. A CNN is a feed-forward network comprised of neurons which are tiled in a way that they respond to overlapping regions in the visual field. It is biologically inspired by how the human eye recognizes changes in the environment, discriminates between objects, and responds to subtle

changes in facial expressions. Each neuron in the brain receives inputs from several other neurons. The effect of each input line on the neuron is controlled by a synaptic weight which adapts so that the whole network learns to perform useful computations. These networks are mainly used for image and video recognition tasks.

A typical CNN is comprised of one or more convolutional layers, one or more pooling layers, and then followed by one or more fully connected layers. There are many convolution kernels in each convolution layer, and each kernel is replicated over the entire image with the same parameters. Each convolution layer is followed by a non-linearity layer which does not affect the receptive filed of the convolution layer. Rectified Linear Units (ReLU) are the most common non-linearity transformation used as they are very fast to train. The pooling layers are mainly important for providing translation invariant features which is critical for object classification and detection tasks. Each layer contains several neurons that are connected to the neurons of the above layer. Layers which are not input or output are called hidden layers. A network learns low-level features in its bottom layers like lines, edges and corners while the upper-level layers learn very complex features. A sample CNN containing only 2 hidden layers is displayed in Figure 1.

Back propagation technique is used to train the CNNs using stochastic gradient descent. The number of parameters to train can range between a few thousands to several millions as the network grows deeper and the size of filters becomes larger. For example, the most popular CNN that was originally designed in [75] contained five convolution layers, 3 fully-connected and 3 max-pooling layers with over 60 million parameters to train (see Figure 2). Although these networks consume several days of training in some cases on GPU machines, the testing only takes fractions of seconds which makes them extremely worthwhile given their significant performance improvement.

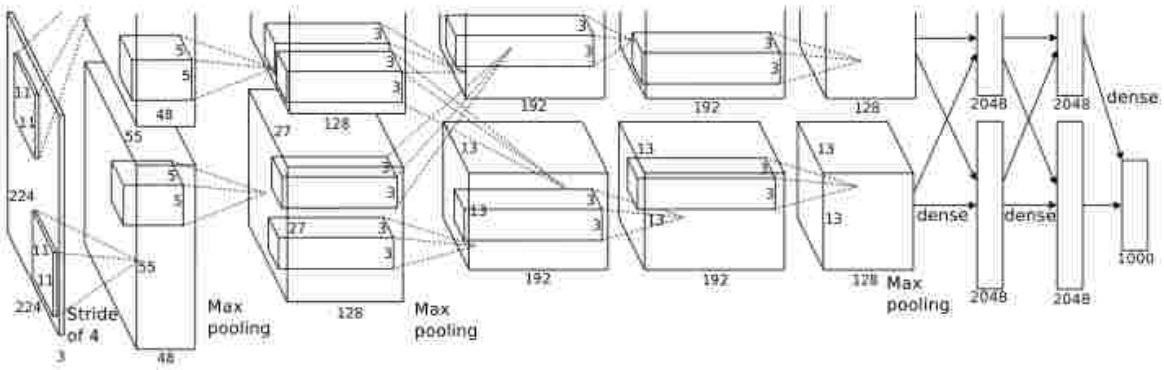


FIG. 2: The famous CNN architecture used by [75] to win the ILSVRC 2012 challenge.

CNNs have recently achieved several record-breaking performances on various benchmark datasets in computer vision. They have achieved an error rate of 0.23 percent on the MNIST database, which is the lowest that have been achieved on the database [32]. A similar performance was achieved on the NORB dataset as well [33]. In the ImageNet Large Scale Visual Recognition (ILSVRC) 2014, which is large-scale visual recognition challenge, almost every highly ranked team used CNN as their basic framework. The wining network consisted of 30 layers achieving a performance close to that of human [116]. Recently, CNNs usage have been extended to image segmentation tasks where it was used to break the performance record on the ISBI 2012 dataset [31].

CHAPTER 3

DEEP MODELS FOR BRAIN EM IMAGE SEGMENTATION

In this chapter, I focused on the automatic segmentation of serial-section Transmitted Electron Microscopy (ssTEM) images. I provided a novel design of deep neural networks (DNN) [78] that extends the techniques described in [31]. I built a pixel DNN classifier that predicts the probability of every individual pixel in a given image being a membrane (border) pixel or not. My DNN classifier accepts raw pixel intensities as input without any preprocessing and learns highly discriminative features automatically before producing final probability maps. These probability maps were fed later to another machine learning classifier based on random forests [20] to produce final segmentations.

3.1 BACKGROUND AND CHALLENGES

Anatomical connections between neurons in the brain form circuits that are responsible for the rapid information flow. Knowledge of the circuit structure is crucial for the investigation of its function. Mapping the structure and components of these circuits is one of the top priority research areas in neuroscience. It provides a foundation for understanding what the brain is made of at the cellular and structural levels, and how these properties change across the normal life span and in brain disorders. The reconstruction of such circuits at a very high resolution using electron microscopy (EM) is considered to be the gold standard for circuit mapping [19].

Currently, sparse circuit reconstruction has been widely used on a small-scale. Most of the studies focused on the very small *C. elegans* or small parts of the nervous systems of the *Drosophila*. Recently, some of those efforts were extended to reconstruct the inner plexiform layer in the mouse retina [22, 57, 73]. Using EM data in large-scale studies is currently a challenge, where the main bottleneck is data analysis. Better automatic image segmentation techniques would substantially amplify the impact of dense EM reconstruction. Machine learning and artificial intelligence approaches are expected to be the main driver for the desired advancements in this area.

DNNs have been widely used in a number of applications in computer vision. It achieved the state-of-the-art performance on tasks like large-scale image and video recognition [66, 75, 133], digit recognition [32], and object recognition tasks [80]. Recently, many attempts have been made to extend the usage of these models to the field of image segmentation, leading to improved performance [21, 61, 62, 103, 121, 135]. Although some of the existing popular models have proved the ability to generalize well for different recognition tasks like the model in [75], most of these models that excelled on image classification and recognition tasks cannot be applied directly to EM image segmentation tasks, given the difference in objectives between those tasks as well as the difference between EM and natural images (more details in section 3.3). As a result, it is desirable to develop an optimized architecture that utilizes the full power of DNNs and tailored specifically for EM image segmentation.

In this work, I developed a DNN model architecture that is highly optimized for ssTEM image segmentation. The key contribution is in the model itself and the novel insights about the specific kernel configuration leading to substantially improved results. I evaluated the effect of model configuration along with kernel structures and depth on the final segmentation outcome. I validated my approach by applying it to the ISBI 2012 EM Segmentation Challenge [59] (`http://brainiac2.mit.edu/`

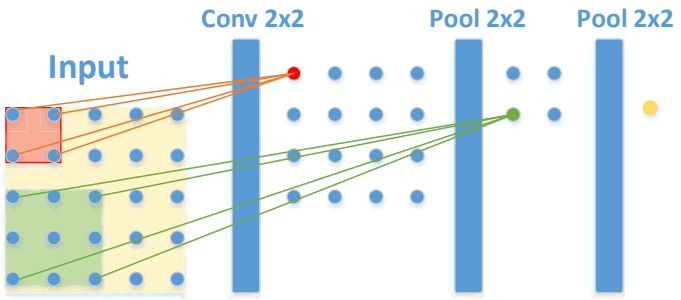


FIG. 3: In this figure, I demonstrate that the upper layers in the network always correspond to a large receptive field due to the presence of max pooling layers. In the figure, even with a shallow network that uses 2x2 convolution for its bottom layer and only 2 max pooling layers, the generated feature map in the last layer corresponds to a receptive field equals to the entire input size.

`isbi_challenge/`), achieving the best performance on all evaluation metrics out of more than 40 participating groups (more groups are participating as the challenge is ongoing). My model was one of the few that were able to beat the performance of a second human observer.

3.2 FRAMEWORK FOR EM IMAGE SEGMENTATION

In this work, I used a full stack of EM image slices of the *Drosophila* first instar larva ventral nerve cord (VNC) [26] provided by the organizers of the ISBI 2012 EM Segmentation Challenge [59]. The training stack consists of 30 grayscale sections of 512 by 512 pixels each, where there is a corresponding label map for each image slice representing whether the pixels are membrane or non-membrane. I trained a deep convolutional neural network (DNN) pixel classifier to predict the label of every pixel

separately. I applied this classifier on another stack of 30 sections representing the testing data where the ground truth is only known to the organizers of the challenge.

In order to use a pixel classifier, I adopted a patch-based training technique. For every pixel in every slice, I extracted a square patch of a fixed size with the target pixel in its center. For boundary pixels, I mirrored the pixels across the slice borders. Upon testing, I obtained a membrane probability map for each slice of the testing data. These probability maps then underwent post-processing to generate the labels.

3.3 ARCHITECTURE AND KEY INSIGHTS

The key contributions of this work are the optimized architecture of a DNN model for EM image segmentation and the underlying motivation and observations. While the use of DNN architecture usually leads to good performance on similar segmentation tasks, a careful design of the network architecture and choices of kernel sizes and placement are the key to utilize the full performance power of the model. For example, in the model suggested by [75], the kernel sizes were chosen to be very large at the bottom layers of the network and then reduced gradually. Later, [133] showed that a better performance could be obtained by reducing the receptive field size and choosing a smaller stride for the first convolutional layer in the same model. On the other hand, in [107], a network of a very small kernel size which was fixed for all layers proved to achieve the best performance on image localization and classification tasks. This highlights the importance of these architectural details in achieving record-breaking performance on different computer vision tasks.

Unfortunately, the application of the very powerful models like [107] and [116] to the task of EM image segmentation does not yield as good results as it did on natural image recognition tasks. The reason is that the two tasks are genuinely different in

terms of their objective and training characteristics. In segmentation, the objective is to assign a label for every single pixel in the image as opposed to a single label assigned to the entire image in classification. In addition, training for segmentation tasks includes a lot of redundancy in the input data due to the patch-based technique, as opposed to training on the whole image or random crops of it in classification tasks. In terms of the data itself, EM images are characterized by their high density and the invariability of the objects it composes unlike the natural images that are regularly used in classification tasks like the ImageNet data [39]. I experimented with several networks that are considered state-of-the-art for image classification and recognition tasks and the results were inferior. In particular, the VGG net [107] has performed very poorly on this specific segmentation task. I attribute this to the very small kernel sizes (3x3) in the network which does not include enough contextual information that this segmentation task requires based on my experiments.

The key observation about this task is how important the context information is for building discriminative features especially in the bottom layers of DNN in EM image segmentation tasks. It is crucial to provide each kernel with a large enough receptive field especially in the bottom layers in order to be able to learn better features. At the same time, if the context is too wide for the bottom layers, the performance will drop due to the excessively large neighborhood which will contain some noise. For example, in the VGG-net [107] all the kernels are fixed to a size of 3x3 which is very small and not enough for the features to be learned from the underlying data. In Krizhevsky's network, they started with an excessively large kernel size of 11x11 at the bottom. When I tried both networks on the EM data, the results were both inferior. In my architecture, I focused on starting with a kernel size of 8x8 at the bottom which is moderately large and sufficient for the network to be able to learn discriminative features without being affected by noise. Increasing

the kernel sizes of the bottom layers in the network increases the receptive fields of each unit in the resulting feature maps, thereby increasing the impact of context information in generating these features. This leads to learning more discriminative features which improves the overall performance. On the contrary, the pixels in the feature maps of the top layers already correspond to a very large receptive field due to the presence of max pooling layers beneath them regardless of the kernel sizes in the layer directly beneath them as illustrated in Figure 3. Using smaller kernel sizes for the upper layers also allows the model to grow deeper as I can add more convolution layers. As a result, I decreased the kernel sizes gradually as I went deeper in the network.

Another key insight is the impact of non-linearity and network depth on the overall network performance. I argue that increasing the number of convolution layers along with their corresponding rectified linear unit (RELU) layers usually increases the network's accuracy. Increasing the number of convolution layers increases the number of features to be learned, while the RELU layers are responsible for increasing the non-linearity in the network and preventing the gradient from saturation. In Krizhevsky network [75], although the input image size was 224x224, the network did not have enough non-linearity with only 5 convolution layers and 5 corresponding RELU layers in between them. On the other hand, my best performing network had 6 convolution layers while the input size is only 95x95. This high non-linearity in my network was crucial for obtaining a better performance. I validated these insights through the network design in the next section.

I used DNN with multiple convolution, pooling and fully-connected layers for the pixel classifier. I experimented with a wide range of window sizes, network depth and kernel sizes to assess the effect of each parameters on the final segmentation outcome. I used window sizes ranging between 35 and 95, depth between 6 and 8 trainable

TABLE 1: The complete architecture of the 4 DNNs used in my experiments.

A	B	C	D
input35x35	input 65x65	input 95x95	input 95x95
conv4-30	conv10-30	conv4-48 LRN	conv8-48 LRN
maxpool			
conv3-50-pad1	conv5-50	conv5-128 conv5-128 LRN	conv6-128 conv6-128 LRN
maxpool			
conv3-60-pad1 conv3-60-pad1	conv3-60 conv3-60	conv6-256 conv6-256 conv6-256	conv4-256 conv3-256 conv3-256
maxpool			
FC-100	FC-100	FC-500 Drop-0.5	FC-500 Drop-0.5
FC-2			
softmax			

layers and kernels between 3x3 up to 10x10.

Although my architectures are quite different from each others, they share some common properties that experimentally proved to be the best for the current task. No preprocessing was applied to any of the networks except for mean subtraction. The overall mean of all the pixels in all sections was subtracted from each pixel value. In my designs, I limited the number of max pooling layers in favor of more convolution layers in all the architectures. Increasing the number of convolution

layers helped the model find more discriminative features. The key challenge was to add as many convolution layers as possible without losing the translation invariance advantage that the max pooling layers provide. In addition, the choice of the window size would always affect the network depth and the number of convolution layers in turn. I compared several architectures containing different numbers of max pooling layers ranging between 2-5 and I found that 3 max pooling layers always give the best results. As a result, each network I trained is divided into 3 blocks; each block contains a number of convolution layers (differs per network configuration and per block) which are followed by a single max pooling layer of size 2x2 and stride 2 by the end of the block.

I also introduced back to back convolution layers interleaved by only RELU as the non-linear transformation in all architectures. Instead of using a single convolution layer with a very large kernel size in every block, I chose to stack multiple convolution layers above each other with moderate kernel sizes while adding RELU layers in between them. This was done mainly to increase the nonlinearity in the model and thus encouraging it to learn more complex features. In addition, breaking down a single large kernel into several smaller ones reduced the total number of parameters need to be trained, thus reducing the overall computation time [107].

My networks concluded with 2 fully connected layers after their third block. The last fully connected layer contains only two neurons corresponding to the segmentation tasks. The outcomes of these two neurons were finally passed through a softmax layer to produce probability values that represent either membrane or non-membrane classes.

I trained 4 DNNs sharing the common characteristics described above but with different configurations. These architectures were inspired by the networks used in [31, 75, 107]. The full architectures of the 4 networks can be found in Table 1.

Networks A and B are both shallow with 6 trainable weight layers each. Network A has a very small window size of 35x35 pixels which highly restricted the depth and the kernel sizes used. Network B starts with a window size of 65x65 pixels which I utilized to test the effect of using excessively large kernels while keeping the depth constant. On the other hand, Networks C and D have a relatively large window size of 95x95 pixels each to include more contextual information and to allow them to grow deep. Both networks use slightly larger kernels than the ones used in network A and slightly less than the ones used in network B.

In my experiments, I hypothesized that starting with a large kernel size in the bottom layers of the network and reducing the size as I move upwards is much better than the opposite direction where I start with small kernels at the bottom. I tested this hypothesis through the configuration of networks C and D. Network C started with a small kernel size for its bottom layer, and then the kernel size was constantly increased till the top convolution layer. For network D, it started by a moderately-large kernel size which is double the initial kernel size of network C then I constantly decreased it. Another advantage of beginning with a large kernel size was that I aggressively reduced the resolution of the feature maps, leading to a reduction in the computations.

To ensure learning in networks C and D, I applied local response normalization (LRN) before the max pooling layers in the first two blocks. This was because the number of parameters to be trained for these networks are higher than the other two due to the larger kernel sizes used and the increasing number of feature maps. Although RELUs are not easily saturated, adding LRNs is supposed to increase model generalization [75]. Experimentally, I found that the performance improvement using LRNs is minimal and does not contribute to the overall performance gain. I still included it in my models for the sake of completeness and to allow reproducibility

of the results. I also applied dropout after the first fully connected layer in both networks C and D (details given in Section 3.5).

3.4 TEST DATA AUGMENTATION

Ensemble learning techniques are well known to improve performance on various learning tasks. Random forest [20] is a clear example that proved to be one of the most effective ensemble learning techniques. In random forests, many decision trees are built based on some random variations in the input and feature spaces, and eventually each decision tree votes for a specific class. The combined vote is then considered as the output decision of the random forest. Extending a similar scheme to neural networks is however computationally expensive. A single DNN usually takes several days of training even using GPU-based implementation for a data set with a million of samples. Training several of these networks would be computationally inefficient.

I chose to perform augmentation upon testing instead of the computationally expensive ensemble of DNNs (similar to the technique used in [34]). I applied several linear transformations on the input image before testing. The transformations were combinations of horizontal and vertical mirroring, and/or rotations by +90, -90 and 180 degrees. After passing the transformed image through the network and obtaining a probability map, a reverse transformation was applied. In total, 8 variations were applied to each testing image and then I took the average. The augmentation was implemented so that each variation received a vote in the final decision produced by the model. This technique is computationally more efficient than training several models as testing time is typically very fast. My experiments showed a considerable advantage of applying those variations as the segmentation error dropped by half as

compared to its original value.

3.5 OVERFITTING REDUCTION

One of the challenges of using a DNN is overfitting. Since my deep networks have a huge number of parameters (up to tens of millions of them), I needed a very large data set to avoid overfitting. In my data set, I extracted all the patches of the membrane pixels in every slice and an equivalent number of non-membrane pixels sampled randomly per slice. This generated about 3 million training patches in total, a number that may not be enough to train a very deep network without the risk of overfitting.

I applied data augmentation to increase the variability in my training data. At the beginning of each epoch, a linear transformation to the input patch was randomly selected. I rotated each patch either by +90 or -90 and/or mirrored it either horizontally or vertically. The data augmentation significantly improved the accuracy of the classifier. In addition, I applied dropout with 0.5 dropping ratio after the first fully connected layer in networks C and D to further improve the performance.

3.6 MODEL DESIGN AND IMPLEMENTATION

My patch-based classifier implementation is based on the publicly available C++ Caffe [68] (branched out in October 2014) with several modifications. I implemented my custom data augmentation where I decoupled cropping from mirroring as it is not desirable in my experiments in addition to implementing rotations with several angles. I also added random shuffling of data at the beginning of each epoch through randomly dropping a few samples in each mini-batch with a specific percentage. A window size can also be supplied before augmentation to determine the patch

TABLE 2: Comparison between the four networks using the validation data set. Both the pixel and warping errors are reported before applying the postprocessing while the rand error is reported after the postprocessing.

Network	Rand error [$.10^{-3}$]	Warping error [$.10^{-6}$]	Pixel error [$.10^{-3}$]
A	82	1548	52
B	95	1729	50
C	75	1775	51
D	47	1684	49

size desired for each configuration to avoid the recalculation of the patches and the databases.

I trained my DNN classifier using back propagation [81] with stochastic gradient decent. I used a mini-batch size of 256, a momentum of 0.9 and a weight decay of 0.0005. I started with a base learning rate of 10^{-2} and decreased it by a factor of 10 every 100K iterations. I used random initialization for the weights through sampling from a normal distribution with a zero mean and 10^{-2} variance. The biases were initialized to either 0 or 1. Experimentally, I found that the model requires 30 epochs of training to achieve the desired accuracy on an NVIDIA K80 GPU. The training time took typically several days to complete.

I used an image-based approach for generating the probability maps for the testing data set. I implemented my own image-based forward propagation code [49]. The model was trained first by Caffe using a patch-based approach then the weights of the kernels and biases were extracted and fed to my image-based code. My image-based

TABLE 3: Comparison between the results obtained using network D before and after using train data augmentation and test data augmentation. These results were obtained by evaluation on the validation data set.

Network	Rand error [$.10^{-3}$]	Warping error [$.10^{-6}$]	Pixel error [$.10^{-3}$]
D	47	1684	49
D - No training augmentation	271	2905	61
D - No testing augmentation	212	2176	61

prediction speeded up the testing time dramatically as compared to a patch-based GPU forward propagation even though my code was running on a CPU (takes roughly 2-4 minutes on a CPU machine). In addition, my image-based code did not require any computational and storage overhead for generating the patches for the testing data, making it much more convenient and efficient.

3.7 POSTPROCESSING

Network D achieved the best pixel accuracy on the ISBI 2012 data set without any postprocessing. However, the best pixel accuracy is not necessarily accompanied with the best segmentation. Even the slightest mis-prediction of certain boundary pixels could severely hurt the overall segmentation. My model is in nature a pixel classifier that was designed to achieve the best pixel accuracy but was not optimized to produce a better segmentation.

I used the watershed merge tree postprocessing technique used in [91, 92]. The technique starts by generating an initial segmentation using watershed for each

TABLE 4: Comparison between my method and the other competing techniques.

Group	Rand error [$.10^{-3}$]	Warping error [$.10^{-6}$]	Pixel error [$.10^{-3}$]
DIVE-SCI (my method)	17	307	58
IDSIA-SCI	18	616	102
optree-idsia	22	807	110
motif	26	426	62
SCI	28	515	63
Image Analysis Lab Freiburg	38	352	61
Connectome	45	478	62
IDSIA-V	46	462	61

probability map and then gradually raising the water level to produce hierarchical segmentations forming a watershed merge tree. A decision of merging two nodes in the merge tree is based on the result of a random forest boundary classifier that predicts the merge based on various features extracted from each two nodes. This scheme reduced the segmentation error to less than half of its original value.

3.8 EXPERIMENTAL EVALUATION

I created a validation data set from the training slices for which the truth labels are available. I divided the 30 training slices into 20 training and 10 validation slices to obtain quantitative evaluations for my models before submitting the final results to the ISBI 2012 challenge. The evaluation of the testing data in this challenge was done through an automated online system where the submitted segmentations were compared to the hidden ground truth based on three different metrics:

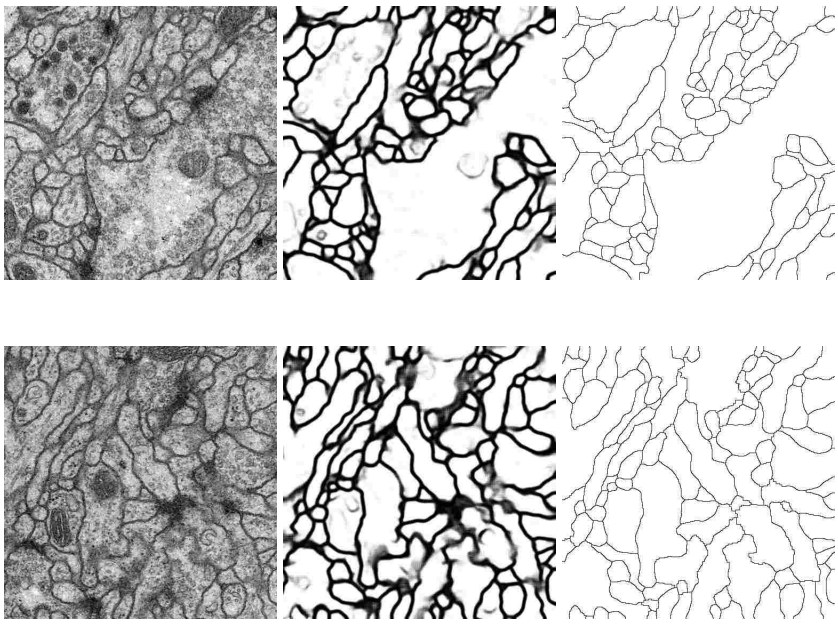


FIG. 4: Slices 1 and 12 from the testing stack are shown along with their segmentations in the top and bottom rows respectively. The first column represents the raw input image, the second column represents the probability map output of the pixel classifier and the third column represents the final segmentation after the post processing.

Minimum Splits and Mergers Warping error is a segmentation metric that penalizes topological disagreements, i.e: the number of splits and mergers required to obtain the desired segmentation.

Foreground-restric Rand error is defined as 1 - the maximal F-score of the foreground-restricted Rand index, a measure of similarity between two segmentations.

Pixel error is defined as 1 - the maximal F-score of pixel similarity, or squared Euclidean distance between the original and the result labels.

TABLE 5: Comparison between the training time and number of parameters for different architectures.

	A	B	C	D
Training hours	5	30	120	104
# of parameters	169,580	196,100	7,615,208	5,719,016

I demonstrate most of the results and model comparisons on the validation data before I show the performance of the best model on the testing data. It is worth mentioning that the results based on the validation data set may be affected by the reduction in the training data set size. This is because I only trained on 20 slices instead of all the 30 slices, since the remaining 10 slices are in the validation set. Nonetheless, the results obtained on the validation data set are very useful for model comparison. Table 2 illustrates the comparison between the four networks evaluated on the validation data set. While all the pixel error values are similar, the rand error gives a better interpretation in terms of segmentation. I observed that the postprocessing reduced the rand error, but increased the pixel and warping errors. This is because the DNN is a pixel classifier aiming at optimizing the pixel error directly, and the warping error is closely related to the pixel error. Thus, I chose to report the pixel and warping errors before I applied the postprocessing. After postprocessing, the pixel and warping errors were highly altered in favor of obtaining a better segmentation. For that, the rand error values reported in Table 2 are obtained after applying the postprocessing.

The results are clearly in favor of the deeper networks C and D, which highlights

the impact of the depth on the outcome of the segmentation. Networks A and B used less number of trainable layers, and this hurt the performance regardless of the kernel sizes used or the model configuration. On the other hand, networks C and D used a large window size with a deeper architecture and relatively large kernel sizes. Network D in particular achieved the best performance among all the other networks in terms of rand index. This validates my hypothesis that starting with a large kernel size in the bottom layers indeed helps the model learn more useful features than starting with a small kernels as in network C.

I report the effectiveness of the techniques described in Sections 3.4 and 3.5 in Table 3. The comparison was made using the best performing network D on the validation data set. I noticed that both the data augmentation and rotations during testing are highly effective for improving the segmentation accuracy. The model suffered from severe overfitting without augmentation while the rotations applied during testing provided robustness to the classifier, thereby reducing the effect of input data randomization to a great extent.

After selecting the best architecture and evaluating different techniques applied upon training and testing, I extended my experiments to the testing data. I trained network D on the 30 training slices and evaluated them using the online system for the ISBI 2012 Challenge. Note that, although winners have been declared before ISBI 2012, the challenge is still ongoing. Figure 9 reports the final results on 2 slices of the testing stack.

On the ISBI 2012 Challenge data set, my approach achieved the best results on all the 3 evaluation metrics, breaking the record maintained by other participants for years. A complete comparison between my approach and other competing methods is shown in Table 7.

In comparison to the second best approach [31] which was also based on a DNN

and uses the same postprocessing technique, my network is deeper and uses a highly optimized network architecture. The largest kernel size used in that network was 5 and the bottom layer starts with a kernel size of 4. They also implemented an actual ensemble of models where they trained several networks with different window sizes and then averaged their results. I believe that applying variations upon testing is much faster and more efficient. The larger kernels utilized in my architecture, the specific configuration of these kernels along with data augmentation techniques I applied are key factors in outperforming all the other competitive approaches in all metrics.

I demonstrate the training time requirements for the different architectures in Table 5. The comparison is based on training on 20 slices of the training stack on a GPU machine. Network C was the most computationally expensive network with the training time reaching roughly 5 days as the number of parameters to train was much higher than the other architectures. However the testing time of all networks was very fast even on my own Matlab-based CPU code with an average of few seconds on network A to roughly 4 minutes on network C.

CHAPTER 4

RESIDUAL DECONVOLUTIONAL NETWORKS FOR DENSE OUTPUT PREDICTION

In this work, I proposed a simple yet powerful model known as residual deconvolutional network (RDN) to address the two conflicting goals of dense output prediction problems: (1) full-resolution prediction and (2) incorporation of sufficient contextual information. My proposed model naturally balances the tradeoff between increasing contextual window required for multi-scale reasoning and the ability to preserve pixel-level resolution and accuracy expected for dense output prediction. I achieved this by adding multiple residual shortcut paths to a fully deconvolutional network with minimum additional computations. This allows for the training of very deep deconvolutional networks that incorporate sufficient contextual information, and the multi-scale full-resolution features are extracted and provided through the residual path. The final dense predictions are made by integrating features computed through both pathways, thereby achieving the conflicting goals in dense output prediction in the same framework.

4.1 DENSE OUTPUT PREDICTION

The automated 3D reconstruction of neurites in brain EM image stacks remains one of the most challenging problems in neuroscience [55, 56, 72]. In such problems, neurons spanning multiple adjacent image slices are expected to be consistently identified and reconstructed. Conventionally, this problem has been approached as a 2D prediction task, where each image slice is segmented individually. Then, a

post-processing step was performed to generate 3D segmentation. The post-processing step usually involves heuristic off-the-shelf classifiers that were trained to link similar segments together across the entire image stack. These classifiers usually rely on hand-crafted features which incorporate the prior knowledge and understanding of the data. Thus, classifiers that worked well on some problems/datasets are not guaranteed to perform similarly in different scenarios. It is thus desirable to design a fully trainable system with minimal post-processing to perform the 3D segmentation task in an end-to-end fashion.

Currently, deep convolutional neural networks (CNNs) [79] are the main drivers used for semantic segmentations. These models are very powerful and highly capable of extracting hierarchical features from image data. They are characterized by their ability to learn features directly from the raw images without relying on any prior knowledge. CNNs have achieved success in different areas of machine learning and computer vision. Improved performance has been achieved in image classification [54, 75, 107, 116, 133] and object detection tasks [105]. Recently, this success has been extended to dense output prediction problems such as semantic image segmentation [43, 93, 98]. These problems find applications in neuroscience of neuronal membrane segmentation in electron microscopy (EM) images [29, 31] and multi-modality infant brain image segmentation [135]. Although deep models are rapidly approaching human-level performance on object recognition tasks, their performance on dense output prediction problems is still far behind human expert performance, especially in brain connectomics involving high-resolution EM image analysis [8, 12, 57, 62, 73, 82, 112, 121].

In dense output prediction tasks such as EM image segmentation, CNNs are expected to generate pixel-level predictions. That is, each pixel in the input image is given a prediction, resulting in a probability map whose size equals to that of

the input image. A common approach to achieve dense prediction is to extract a fixed-sized patch centered on each pixel and employ a regular CNN as used in image classification to determine the label of the center pixel [31, 42]. However, such approaches only incorporate limited contextual information contained in the patch. Contextual information can be increased by enlarging the patch size, but excessively large patches tend to compromise the full-resolution, pixel-level predictions. Thus, dense output prediction problems face the conflicting goals of full-resolution prediction and incorporation of sufficient contextual information [129].

4.2 RESIDUAL DECONVOLUTIONAL NETWORKS

Most of the dense prediction methods do not explicitly address the problem of losing pixel-level resolution. This is mainly because most of the CNNs that were used for dense output prediction are variations of the ones that achieved excellent performance on classification and recognition tasks. In those tasks, it is a common approach to reduce the feature map sizes using pooling layers to increase the receptive fields of the resulting feature maps, thereby increasing the contextual window used to generate the single prediction for a given image. When those networks are tailored towards dense prediction, the attempts to reconstruct a full-resolution prediction is hampered by the loss of pixel-specific resolution information.

Fully convolutional networks (FCNs) [30, 93] are efficient approaches to generate dense predictions for image segmentation. The idea is to reconstruct the full-sized input by performing several deconvolution operations at multiple scales through aggregated bilinear interpolation. The segmentation performance of FCNs is limited by the absence of real deconvolution, and full-resolution features are not well preserved. To address this limitation, deconvolution networks [98] have been proposed recently by

performing actual deconvolution. The pooling layers are reversed in the decoding stage by unpooling layers which keep track of the maximum activation position selected during the pooling operation. While both of these two approaches are attempts to design novel deep models specifically for dense prediction problems, they do not have explicit mechanisms to address the conflicting goals in dense prediction problems. They still suffer from loss of information due to excessive reduction of resolution as I show in my experiments.

In the design of my model, I intend to achieve three goals: (1) Generate dense predictions equal in size to any arbitrary-sized input. (2) Increase the receptive fields of output maps to increase the contextual information used to make pixel-level decisions. (3) Achieve pixel-level accuracy by incorporating high resolution feature information.

I build on the deconvolution scheme proposed in [98] to generate dense predictions. I enhance the performance of deconvolution networks by adding residual connections between every several stacks of convolution or deconvolution layers. These shortcut connections perform projection mapping and are added to the output of the stacked layers with minimum additional computation cost. It has been shown in [54] that it is much easier to optimize a residual mapping (with shortcut connections added) rather than the original plain one. Residual networks in [54] also demonstrated a significant performance gain as a result of increased network depth on tasks of image classification and object detection. For my dense prediction network architecture, I propose to introduce projection shortcuts not just on the convolutional stage responsible for extracting the feature representations, but also on the deconvolutional stage responsible for reconstructing the shape and producing the objects segmentation. I believe that with this design, my network is able to acquire more multi-scale contextual information while reducing the effect of the degradation problem [54, 110].

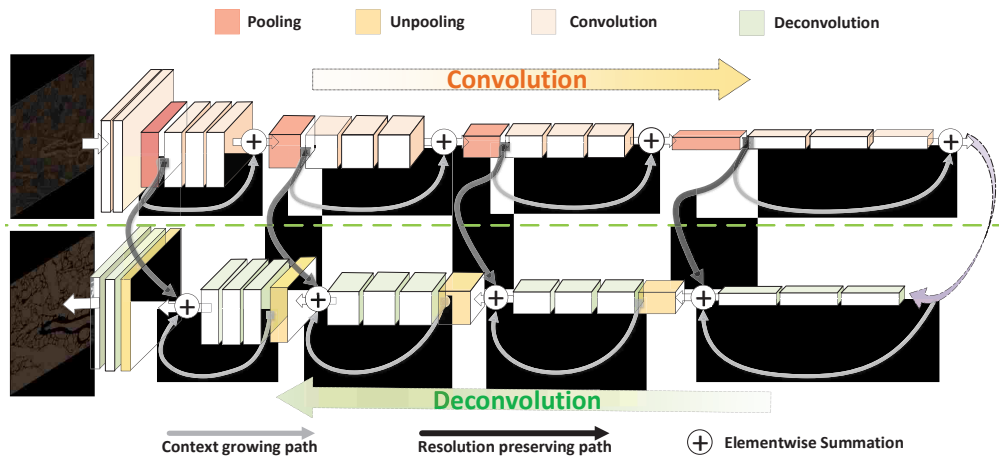


FIG. 5: Architecture of the residual deconvolutional network (RDN). The network consists of two pathways, namely context-growing path and resolution preserving path. All convolution and deconvolution layers in the encoding and decoding stages are of size 3x3. A kernel of size 1x1 is used to implement the projection mappings. Max pooling is used to reduce the feature map sizes in the convolution stage while unpooling is used to restore the original size in the decoding stage.

I also propose the use of a novel resolution-preserving path to facilitate the reconstruction of full-resolution output. The resolution-preserving paths are essentially the projection mapping of the pooling layer outputs added to the output of the corresponding deconvolution layer before performing the unpooling operation. These paths are responsible for transferring the missing high resolution information from the encoding stage to the decoding stages. Together, the context-growing and the resolution-preserving paths have significantly boosted the performance of non-residual deconvolutional networks as shown in Section 4.6. An illustration of the RDN

architecture is shown in Figure 5.

4.3 NETWORK ARCHITECTURE AND TRAINING

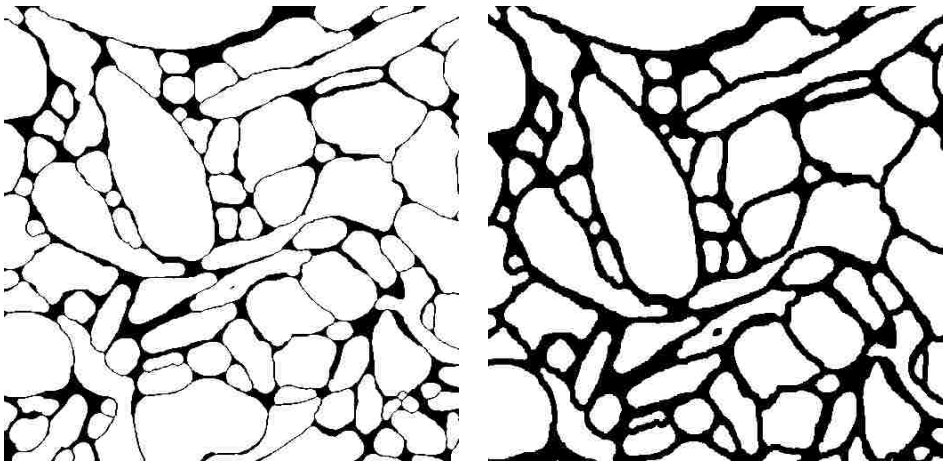
The RDN architecture is mainly inspired by the ideas in [54, 98, 107] with three main differences:

- The convolution stage of the network has been mirrored in the deconvolution stage to produce dense probability maps instead of a single value prediction for each training instance. Context-growing paths have been added to the deconvolution as well as the convolution layers. Also, resolution-preserving paths have been added to transfer resolution-specific information from encoding to decoding stages.

- The input to the network is 3D patches extracted from consecutive slices to exploit the 3D aspect of the data in a way similar to how a human annotator perform segmentation. Square patches were extracted randomly from the entire image stack. I performed mirror-padding for patches extracted from the first and last slices to generate the 3D input to my network.

- Use of 3D kernels in the early stages of the network to learn cross-sectional features from the 3D patches provided as an input to the network.

The network contains 23 convolutional layers and 20 deconvolutional layers in total as the network is not entirely symmetric. The kernel sizes are either 3x3 or 1x1 when I performed branching before adding the residual paths. Zero padding was used whenever size preserving was needed in the learned layers. I added a batch normalization layer after each learned layer and rectified linear units were used as



Original label image Widened border label image

FIG. 6: An illustration of the effect of border widening on the training labels. I show the original label image (slice 50 of the training stack) on the left and the corresponding altered label on the right. I show that label widening reduces the segments sizes and increases the distance between them.

the non-linearity transformation. I used a patch size of 128x128x3 in training while the entire image was used in testing.

No pre-processing was used on the raw input images. However, I modified the training labels to reduce the segment sizes by increasing the border width in-between them (see Figure 6. The widening of borders was done using a minimum kernel of size 5x5 by assuming that all segments are having a label of 1 and borders are having a label of 0 in the ground truth label stack. Any pixel that was in a neighborhood of size 5x5 of a border pixel was considered to be border as well. Label widening was crucial in allowing the network to differentiate border from non-border pixels.

My model implementation was based on the publicly available C++ Caffe [68]. I trained the RDN using back propagation [79] with stochastic gradient decent. The mini-batch size used was 15 as the dense prediction requires a lot of memory. However, the network requires roughly 15k iterations to achieve its full potential due to the existence of residual paths which speeds up the computations. I used a momentum of 0.9 and weight decay of 0.005. I started with a base learning rate of 10^{-2} with a polynomial decay. Random initialization was used for all learned layers. The experiments were carried out on an NVIDIA K80 GPU machine, taking roughly 2 days of training.

To improve the robustness of the resulting probability maps, I applied 8 variations to the testing images before passing them down through the network. A reverse transformation was then applied to each resulting probability map before taking the average across all variations. The transformations were combinations of horizontal and vertical mirroring, and/or rotations by +90, -90 and 180 degrees.

4.4 EM IMAGE DENSE PREDICTION PROBLEM

In my experiments, I used two separate datasets [70] for training and testing from the ISBI 2013 challenge. Each dataset is a 3D stack of 100 sections from a serial section scanning electron microscopy (ssSEM) of mouse cortex. The pixel resolution is 6x6x30 nm/pixel which covers a microcube of approximately 6x6x3 microns. Both datasets have high x- and y-direction resolution whereas the resolution of z-direction is low. The neurites in the training stack have been manually delineated, generating a corresponding label stack of 100 sections. The training stack contains 400 neurites that have been labeled consistently across the 100 slices. Some neurites are split into several segments in some slices while still required to preserve their unique label

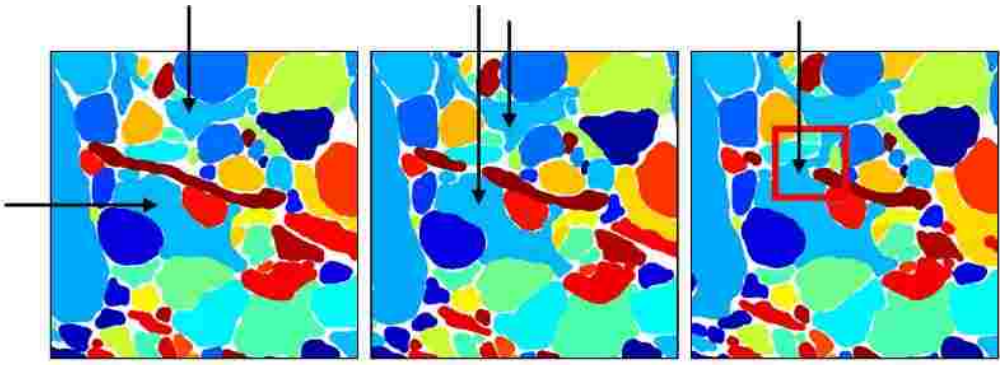


FIG. 7: In this figure, I show the images of the first 3 consecutive slices of the training data cropped at the same position. Segments having the same color across the 3 slices represent the same neurite. The arrows point towards a neurite that has been split in slices 1 and 2 while it appears as a single segment in slice 3. The required segmentation should assign the same label for the splits in slices 1 and 2 even if they are not connected as they belong to the same neurite.

across sections, which increases the complexity of the 3D segmentation task (see Figure 7). The labels of the testing stack are not available to challenge participants.

I formulated the 3D segmentation problem of 400 neurites in the training stack as a single 2D segmentation problem. I built a pixel classifier (Section 4.2) that accepts patches extracted from the raw input image to generate 2D probability maps. Each resulting probability map indicates the probability of each pixel being either a foreground (neurite) pixel or background (border). The probability maps have no reference of which neurite a pixel belongs to, had it been identified as a non-membrane pixel. The final 3D segmentation was obtained by a simple post-processing technique described below.

4.5 POST-PROCESSING

Super-pixel level algorithms are commonly used as a building block in most post-processing techniques for 2D and 3D segmentation tasks [43, 63, 92, 138]. They are used mainly to generate an over-segmentation from probability maps or affinity graphs. Later, another classifier is built on top of the results of the super-pixel level algorithms to accurately merge some of the overly segmented regions. The key limitation of these approaches is that they reduce the generality of the overall proposed techniques, since they rely on hand-crafted features to build classifiers on top of super-pixel algorithms. One of the fundamental advantages of the proposed method is the ability to learn features from the data, hence their ability to generalize to many other datasets.

It has been shown before [43] that relying heavily on the learned network while simplifying post-processing could result in a dramatic increase in the speed of computations while maintaining the generalization of the proposed technique. I followed this scheme by applying 3D watershed algorithm directly to the entire probability map stack to generate the final segmentations. The 3D watershed method uses 26-connected neighborhoods to determine the catchment basins in an image. I blurred the probability maps with a Gaussian kernel of size 6x6 and a standard deviation of 1. I also suppressed all minima in the probability maps whose depth were less than a specific threshold. This threshold was mainly used to control the level of over-segmentation and can be tuned using the training data. The model tends to reduce the predicted segment sizes due to the widening of training label borders described in Section 4.3. As a result, I applied a reverse transformation which used a maximum kernel to increase segments sizes. The overall processing is simple, fast and requires minimum additional computations.

The 3D watershed method does not rely on any hand-crafted features and needs only 1 parameter to be tuned. The quality of the probability maps generated by the RDN is a key for the 3D watershed to be able to generate the final segmentations directly without relying on any additional computations. I demonstrate in Section 4.6 that its performance on probability maps with lower quality is hampered, making the use of more sophisticated post-processing techniques a necessity.

4.6 RDN MODEL EVALUATION

In my experiments, I divided the training stack into 80 slices for training and the rest for validation. I trained the network on the training data for 15K iterations using random sampling of patches. I then evaluated on the validation data. The adapted Rand error metric was used by the ISBI 2013 challenge organizers [60] to assess the segmentation results. The adapted Rand error is defined as: 1 - the maximal F-score of the Rand index (excluding the zero component of the original labels). To evaluate on the testing data, I relied on an automated online system where my submitted results were compared to the hidden ground truth labels available only to the challenge organizers.

I demonstrated the superiority of my proposed model over other techniques by using the results in Table 6. The results were obtained by applying the 3D watershed method to the validation probability maps as discussed in Section 4.5. I performed grid search to obtain the best parameter to control the over-segmentations for each set of probability maps independently while applying 3D watershed. I compared my proposed RDN to three other techniques:

- IDSIA [31]. Probability maps obtained from training a CNN. These probability maps are provided by the challenge organizers as an optional parameter to use by

TABLE 6: Comparison between different techniques applied to the validation data. The performance reported is after applying 3D watershed with the best over-segmentation threshold for each set of probability maps independently.

Method	Rand error
RDN	0.0814
IDSIA	0.1184
Deconvolution Network (DN)	0.1514
DIVE CNN	0.1541

the participants to evaluate their proposed post-processing techniques.

- DIVE CNN [42]. Probability maps obtained from my CNN model proposed in Chapter 3. I used the same model to generate the probability maps for the 3D challenge data.

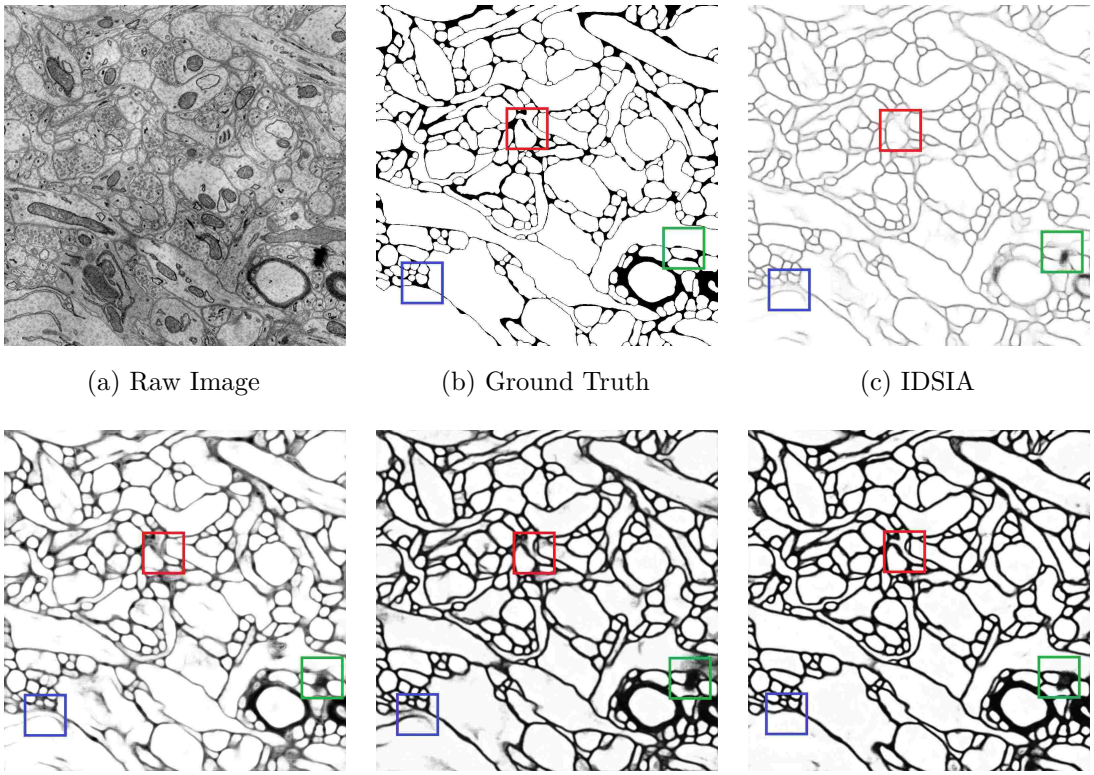
- Deconvolution Network (DN): Probability maps obtained by training the same exact RDN network without the residual paths.

From Table 6, it is clear that my RDN outperforms all the other CNN-based models by a significant margin. For the IDSIA probability maps, I do not know which slices were used as validation data by the generating group as they provided their probability maps for the entire training stack. However, I assumed that they used the same validation slices as mine (slices 1-20 from the training stack). This assumption is either fair or in favor of the IDSIA probability maps in case the chosen slices were in fact used as training instances by them. Nonetheless, my RDN still achieved a much better segmentation using 3D watershed. A qualitative evaluation of

the performance of those models are provided in Figure 8. I showed the probability map generated for the same slice by different networks and I highlighted sample areas of improvement in colored boxes. My RDN was able to recover most of the missed borders by the CNN trained by IDSIA and DIVE and also improve the certainty of some others. In contrast to DN, my RDN is less sensitive to noise and produces clear probability maps. I compare the final segmentation obtained from my RDN, the IDSIA CNN and the DIVE CNN in Figure 9 after applying 3D watershed. I showed 3 consecutive slices from the validation data where pixels have been consistently given the same color across the 3 slices to denote that they belong to the same neurite. I noticed that the quality of the probability maps generated by RDN has significantly impacted the segmentation results. The CNN-based probability maps result in poor segmentation by either splitting or merging many segments, thus requiring the use of more sophisticated post-processing methods.

I applied my trained model on the testing stack where the labels are hidden and submitted the results to the ISBI 2013 challenge. I achieved the 2nd ranking among many participating groups. Most of the challenge participating groups are working on improving post-processing techniques while relying entirely on the probability maps provided by the IDSIA group. For example, the leading group generated over-segmentations based on the IDSIA probability maps and then built a random forest classifier based on features computed from the over-segmentations. In contrast to these technique, I do not rely on any hand-crafted features throughout the entire processing pipeline and my method is very fast with minimum additional computations. The corresponding team rankings are shown in Table 7. I note that this challenge is an ongoing one and rankings are subject to change as more teams start joining.

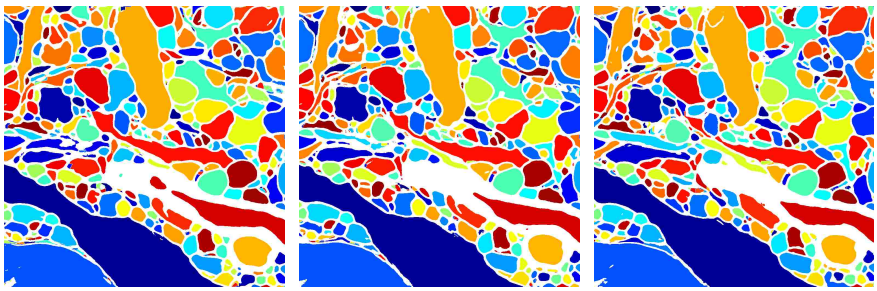
By analyzing the provided dataset, I noticed that the testing stack contains segments with much larger sizes than the ones present in the training stack and



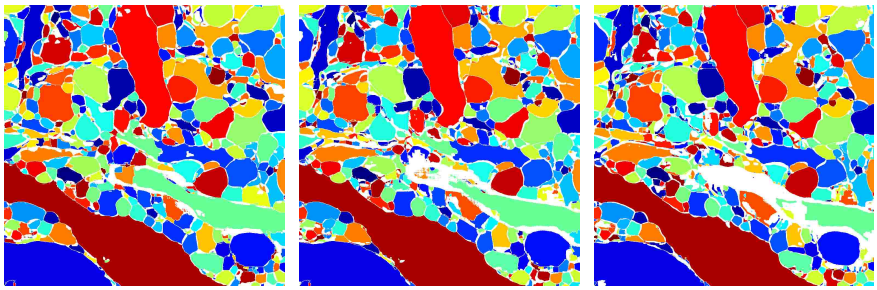
(a) Raw Image (b) Ground Truth (c) IDSIA

(d) DIVE CNN (e) DN (f) RDN

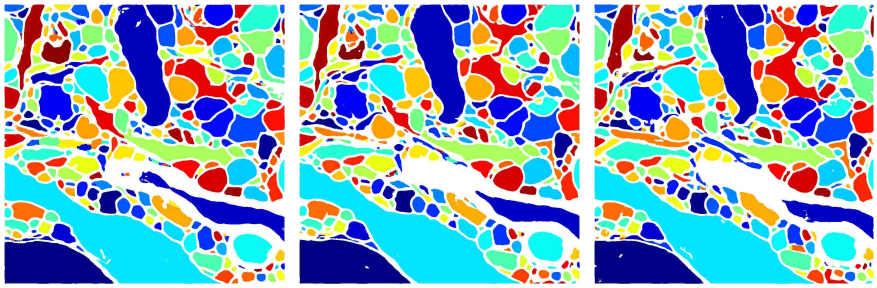
FIG. 8: Qualitative comparison between the results of several models with respect to training slice number 50. Sample areas with clear differences are marked with colored boxes. A: Raw input image. B: Ground truth 2D label image. C: Probability map generated by group IDSIA with a CNN [31]. D: Probability map generated by a CNN 3. E: Probability map generated by a deconvolution network (DN). F: Probability map generated by my RDN.



RDN



IDSIA



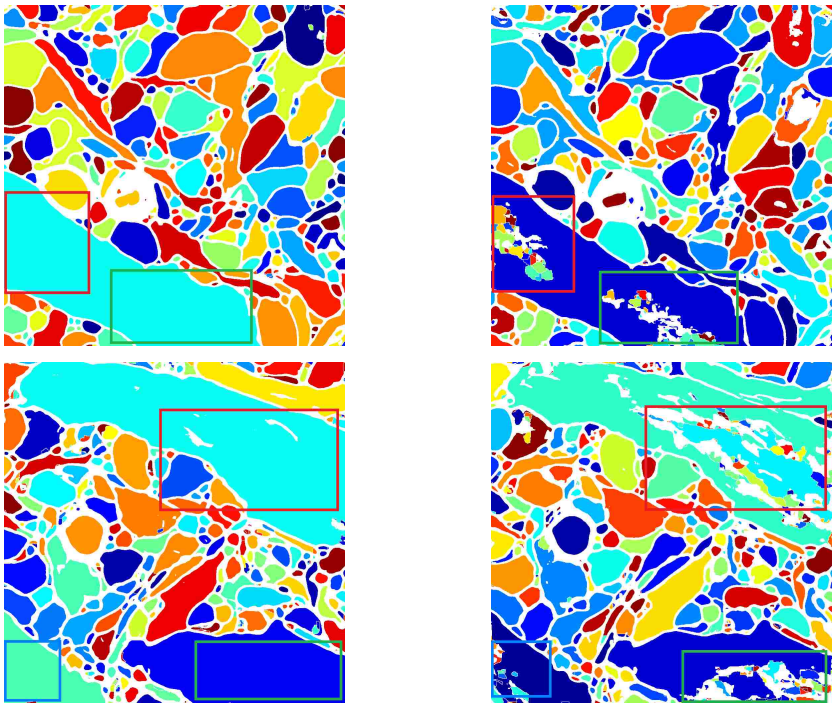
DIVE

FIG. 9: Comparison between the segmentations obtained by applying the 3D watershed method to different networks. I applied it to my RDN probability maps (top row), the IDSIA probability maps (middle row) and the DIVE probability maps (bottom row). I show 3 consecutive slices from the training stack (slices 2-4) to demonstrate that pixels belonging to the same neurite are segmented consistently (identified by the same color across slices).

TABLE 7: Comparison between my method and the other techniques in the ISBI 2013 challenge. This ranking is based on the results published on the challenge leaders board on May 19, 2016. I showed only the top 9 teams.

Group	Rand error
IAL	0.07107
DIVE (my method)	0.09104
Team Gala	0.10041
SCI [92]	0.10829
MIT [138]	0.11361
Anonymous	0.11501
FlyEM [99]	0.12504
rll	0.13111
Rhoana	0.14835

with a higher frequency. As a result, a regular deconvolution network is not able to recognize those large segments, resulting in their over-segmentations. I highlight the effectiveness of the proposed RDN model in dealing with this problem and the importance of resolution-preserving paths in Figure 10. I trained a RDN model without resolution-preserving paths and evaluated it on the testing stack where this problem occurs. I noticed that without resolution-preserving paths, the network was not able to reconstruct the full-resolution output effectively, resulting in a poor over-segmentation of very large segments. On the other hand, a regular RDN avoided this over-segmentation, thereby confirming its ability to reconstruct full-resolution output by using the resolution-preserving paths.



With resolution-preserving paths Without resolution-preserving paths

FIG. 10: An illustration of the effect of resolution-preserving paths on the final segmentation. Colored boxed have been placed on the compared segments.

4.7 VALIDATING MODEL GENERALITY

To demonstrate the ability of my proposed model to generalize, I extended my experiments to the ISBI 2012 dataset [26]. The dataset consists of a full stack of EM image slices of *Drosophila* first instar larva ventral nerve cord (VNC). The stack contains 30 grayscale sections of 512 by 512 pixels each. In my experiments, I divided this stack to 20 training slices and 10 for validation. I trained the same RDN classifier

TABLE 8: Comparison between my RDN and the DIVE CNN segmentations on the ISBI 2012 challenge validation set.

Method	Rand error	Warping error	Pixel error
RDN	0.0282	0.0026	0.0937
DIVE CNN	0.0388	0.0029	0.0939

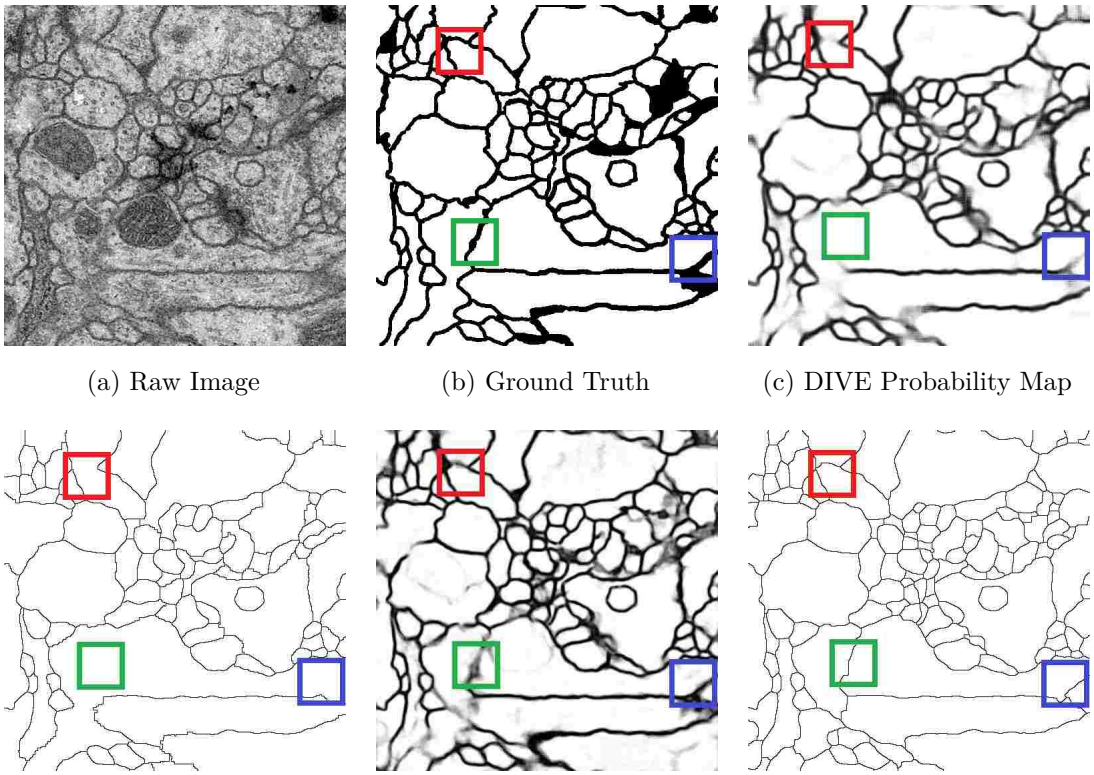
explained in Section 4.3 with only a few differences:

- 2D patches were extracted instead of 3D. This is mainly due to the extremely low Z-direction resolution for the provided data.
- Only 2D kernels were used through the entire network.

I used 2D watershed as my post-processing with only 8 neighborhood pixels used to determine the catchment basins in the image. Again I did not rely on any problem-specific post-processing technique to ensure the generalization of my technique. I compared my results with a CNN-based classifier trained by the DIVE group [42] participating in the ISBI 2012 EM segmentation challenge [59]. They used an advanced post-processing technique [91, 92], where a random forest classifier is built on top of super pixels output followed by building a Merge Tree (MT). Unlike my technique, the features extracted for the random forest classifier in the MT are generated based on prior knowledge of the data.

I used three common metrics to evaluate the segmentations generated:

Minimum Splits and Mergers Warping error is a segmentation metric that penalizes topological disagreements, i.e: the number of splits and mergers required to



(a) Raw Image (b) Ground Truth (c) DIVE Probability Map

(d) DIVE Segmentation (e) RDN Probability Map (f) RDN Segmentation

FIG. 11: Qualitative comparison between the probability maps and segmentations obtained from my RDN and the DIVE CNN based on slice 21 from the training stack. Sample areas with clear differences are marked with colored boxes. A: Raw input image. B: Ground truth 2D label image. C: Probability map generated by the CNN proposed in 3. D: Segmentation generated by CNN and MT. E: Probability map generated by my RDN. F: Segmentation generated by my RDN and watershed.

obtain the desired segmentation.

Foreground-restric Rand error is defined as 1 - the maximal F-score of the foreground-restricted Rand index, a measure of similarity between two segmentations.

Pixel error is defined as 1 - the maximal F-score of pixel similarity, or squared Euclidean distance between the original and the result labels.

I compared the results of my RDN followed by 2D watershed to the DIVE CNN followed by Merge Tree in Table 8. The results are obtained from evaluating on the validation data (slices 21-30). My RDN with a general post-processing technique clearly outperforms its CNN counterpart across all evaluation metrics. I note that the Rand error is believed to be the most suitable metric to evaluate semantic segmentations as it penalizes over and under segmentations of objects instead of pixel mispredictions. The improvement obtained from using my RDN is mainly demonstrated in the improved Rand error value. I provided a qualitative comparison between both techniques in Figure 11.

4.8 SUMMARY OF CONTRIBUTIONS

I evaluated my proposed method on the challenging problem of neurites segmentation from 3D EM images, which is a key step in dense brain circuit reconstruction. I participated in the open challenge on 3D EM image segmentation [60], and I achieved the second place among many teams. Note that most of the challenge participants rely on given probability maps generated by a CNN as inputs to their techniques and focus on creating heuristic post-processing techniques to generate final segmentations. In contrast, I used end-to-end trainable models with minimum post-processing to achieve the top results. My technique does not rely on prior knowledge of the data.

I thus expect my method to generalize well to other dense output prediction problems. I demonstrated this by extending my experiments to the ISBI 2012 challenge dataset [59], where I achieved consistently high performance. The results generated by my method can be coupled with any post-processing technique used in the challenge, leading to improved performance.

CHAPTER 5

INTEGRATIVE REGIONAL ANALYSIS OF THE GENOME AND CONNECTOME

In this work, I study the relationship between gene expression and brain connectivity in a single rodent brain, namely the mouse brain. My investigation is made possible by the recent release of the mouse brain connectivity data from the Allen Mouse Brain Connectivity Atlas [2]. By integrating this resource with the Allen Mouse Brain Atlas data [3, 84], I attempt to systematically study the relationship between gene expression and brain connectivity in a single mammalian brain. I employ ensemble learning methods [137] for predicting the brain connectivity using gene expression data. These methods generate many base models by randomly sampling the original training data, thereby yielding accurate and robust predictions [47, 52, 90, 132]. I consider two types of base models in this work, that is, tree and sparse models, which have been commonly used in neurological applications [35, 38, 47, 52, 104, 128]. One common and appealing property of these models is that they can perform feature selection and prediction simultaneously, thereby enabling us to identify genes that retain high predictive power.

5.1 RELATIONSHIP BETWEEN GENOME AND CONNECTOME

The mammalian brain contains a large number of cells connected into an interaction network that controls the information flow among neurons [114, 124, 125]. The brain connectome plays a pivotal role in generating the cognition, emotion, and perception of an organism. Neurological diseases, such as autism and schizophrenia,

are commonly found to be associated with abnormal brain connectivity [48, 69, 77]. Hence, understanding the brain functional circuitry becomes one of the central research themes in neuroscience. At the cellular level, each neuron is largely unique in the sense that it contains a unique combination of proteins that determine how the neuron functions. At the molecular level, the proteins in a neuron are encoded by the genome, which also contains regulatory sequences to control when and where each gene is turned on or off at what level. In other words, the fundamental biochemistry of neurons is determined by spatiotemporal gene expression and regulation encoded into the genomic regulatory networks. This prompts research efforts to characterize the cellular localization of gene expression in the brain and investigate the relationship between genome and connectome [13, 16, 27, 41, 64, 85, 115, 117, 119, 136].

The initial attempts to investigate the relationship between gene expression and neuronal connectivity focused on the nervous system of *C. elegans*, because the synaptic connectivity in this organism is known. In [123], computational techniques were presented to link gene expression and neuronal connectivity. In addition, sets of synergistically interacting genes were identified based on entropy minimization and boolean parsimony. Experimental results showed that the synergistic expressions of a subset of genes are predictive of neuronal connectivity. [71] used correlation and prediction analysis assays to study the relationship between gene expression and neuronal connectivity in *C. elegans*. They showed that the expression signature of a neuron carries significant information about its synaptic connectivity. They also identified a list of putative genes that retain high predictive power. [11] studied the molecular markers and logic that direct synapse formation in *C. elegans*. They built a probabilistic model and attempted to explain the neuronal connectivity diagram of *C. elegans* as a function of the expression patterns of its neurons. Their results showed that the synaptic connections in *C. elegans* can be predicted by using the

expression patterns of only a small number of genes.

Motivated by prior research results on *C. elegans*, a few recent studies have attempted to investigate the relationship between gene expression and brain connectivity in the rodent brain. Since the gene expression and brain connectivity data were not available in a single rodent species when those studies were performed, they usually fused data from two different species [44, 127]. Specifically, [44] used the gene expression data of the mouse brain from the Allen Brain Atlas [113] and the connectivity data of the rat brain from the Brain Architecture Management System [15] to study the relationship between gene expression and brain connectivity. By using a series of covariation analysis techniques, they reported that gene expressions in the mouse brain are correlated to the connectivity in the rat brain. In addition, they identified a set of genes that are most correlated with connectivity. [127] used the same sets of data and tried to predict regional connectivity in the rat brain by using gene expression data from the mouse brain. They also identified a set of highly predictive genes whose functional roles in disease conditions were evaluated.

My experimental results show that gene expression is predictive of connectivity in the mouse brain when the connectivity signals are discretized. When the expression patterns of 4084 genes are used, I obtain a predictive accuracy of 93%. My results also show that the expression patterns for a small number of genes can almost give the full predictive power of using thousands of genes. I can achieve a prediction accuracy of 91% by using the expression patterns of only 25 genes. Gene ontology analysis of the highly ranked genes shows that they are significantly enriched for connectivity related processes. I also performed covariation analysis on the gene expression and connectivity data. My results show that gene expression and connectivity are correlated in the mouse brain. I show that my results on prediction and covariation analysis are significant when the spatial autocorrelation effects are considered.

5.2 DATA ATLASES AND PROCESSING

The Allen Mouse Brain Connectivity Atlas (the Connectivity Atlas) provides 3-D, high-resolution maps of neural connections in the adult mouse brain [2]. In this atlas, axonal projections mapped from major anatomical regions are labeled by recombinant adeno-associated virus tracers and visualized using serial two-photon tomography. The primary data consist of high-resolution images that capture the axonal projections from anatomic regions throughout the brain. The sectional data set from each experiment is associated with a primary injection structure and possibly multiple secondary injection structures. To facilitate the integrative analysis of the connectivity atlas and other mouse brain atlases such as the gene expression atlas, each image series is registered into the Allen Mouse Brain Reference Atlas (the Reference Atlas) and converted into grid-level voxel data by an informatics data processing pipeline [5].

The Allen Mouse Brain Atlas (the Gene Expression Atlas) provides genome-wide, 3-D, high-resolution *in situ* hybridization (ISH) gene expression data for approximately 20,000 genes in the sagittal section for the 56-day old male mice [3]. In addition, coronal sections at a higher resolution are available for a set of about 4,000 genes showing restricted expression patterns. For each experiment, a set of high-resolution 2-D image series are generated, and they are subsequently processed by an informatics data processing pipeline to generate grid-level voxel data in the Reference Atlas space [4, 97]. The output of the pipeline is quantified expression values at a grid voxel level and a structure level according to the Reference Atlas ontology [40].

The registration of the Connectivity Atlas and Gene Expression Atlas data to the same reference space enables the integrative analysis of these two types of data. I retrieve the Gene Expression Atlas ISH expression energy grid data, the Connectivity

Atlas grid data, and the Reference Atlas ontology and annotation data from the Allen Brain Atlas API [1]. I use the coronal section data set, since genes in this data set show restricted expression patterns, and the data have higher resolution. This set consists of 4347 genes, and 4084 of them contain nonzero expression energy in the brain structures I am studying. Hence, the total number of genes I used is 4084. For both the gene expression and the connectivity data, I apply the structure unionizer module to compute the signal statistics for each structure by combining grid voxels with the same 3-D structural label. Note that the reference atlas ontology is organized as a hierarchy, and the reference atlas is typically annotated at the lowest level of the hierarchy. Hence, statistics at upper level structures can be obtained by combining measurements of the hierarchical children.

5.3 BRAIN CONNECTIVITY PREDICTION

In this work, I aim at an integrative analysis of the gene expression and connectivity atlases in the mouse brain. I derive structure-level gene expression and connectivity data from the Gene Expression Atlas and the Connectivity Atlas, respectively. The gene expression data is used to construct the feature matrix, and the connectivity data is used to construct the label matrix. Then computational models are built to predict the output labels using the input features. I formulate the prediction of connectivity from one structure to all other structures as a two-class classification task. In each task, the positive examples are the gene expression vectors for structures that are connected from the current structure; the negative examples are the gene expression vectors for all other structures. This formulation is illustrated in Figure 12.

While the Reference Atlas contains over 800 brain structures, the release of the Connectivity Atlas I use (October, 2012) provides connectivity data that include

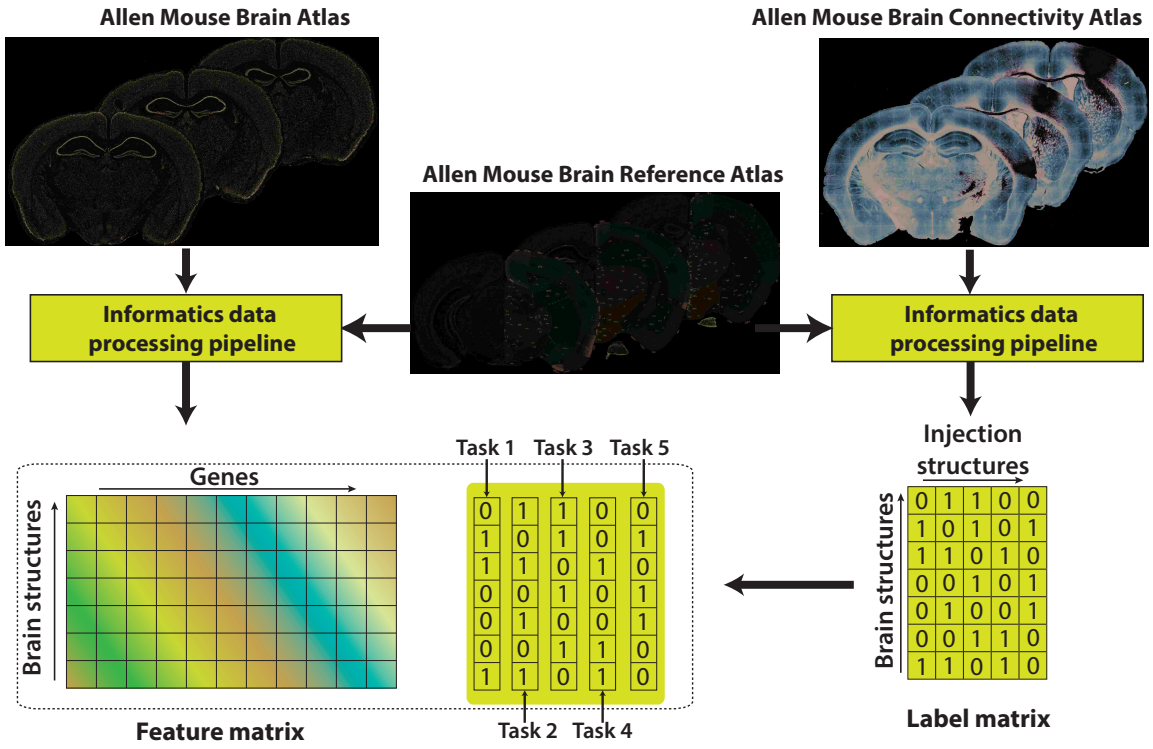


FIG. 12: Illustration of the proposed brain connectivity prediction framework. The feature matrix is derived from the ISH gene expression data in the Allen Mouse Brain Atlas, and the label matrix is based on the connectivity data in the Allen Mouse Brain Connectivity Atlas. Computational models are trained to predict the labels (connectivity) using the features (gene expression). In this example, I have gene expression data for 7 structures and 11 genes that correspond to the rows and columns, respectively, of the feature matrix. I also have connectivity data from 5 experiments corresponding to 5 injection structures (columns of the label matrix). The connectivity information from these 5 structures to the 7 structures are encoded into the label matrix. I formulate 5 classification tasks where each task models the connectivity from one of the 5 injection structures to the 7 structures. The binary connectivity data are obtained by thresholding the structure level connectivity data, and the thresholds are selected so that the numbers of positive and negative samples in each task are approximately balanced. Detailed explanations are given in the text.

160 structures as the primary injection structure and 301 structures in both the primary and the secondary injection structures. I compile two data sets, i.e., primary injection driven (PID) and all injection driven (AID), based on how I aggregate the connectivity grid data sets. In compiling the PID data set, I ignore the secondary injection structures and aggregate all grid data with the same primary injection structure by computing voxel-level average. The aggregated grid data are considered to encode the connectivity information from the primary injection structure to all other brain voxels. In contrast to the PID data set, I consider both primary and secondary injection structures in compiling the AID data set. Specifically, for each of the 301 brain structures that appear as either primary or secondary injection structures, I aggregate all grid data that include this structure as either primary or secondary injection structures. Similarly, the aggregated grid data are considered to encode the connectivity information from this structure to all other brain voxels. I compute the structure-level connectivity data by applying the structure unionizer module to the aggregated connectivity grid data.

I initially attempted to treat all structures that have a nonzero connectivity signal from the current structure as positive examples, but I discovered that this scheme usually leads to highly unbalanced numbers of positive and negative examples. In particular, the numbers of negative samples are zero in a significant number of tasks. Overall, only 8.5% of the class labels in the PID data and 5.4% of the class labels in the AID data are zero. I observe that this is primarily due to the way I aggregate voxel-level connectivity signals into structure-level information. Specifically, when any voxel in a structure gives nonzero signal, the connectivity to this structure becomes nonzero. Typically, each structure contains a large number of voxels, and thus the data aggregation scheme is sensitive to false positive signals introduced into the data during experiments and informatics processing.

Since it is unlikely that many brain structures are connected to most or all other structures, I discretize the connectivity signal by applying nonzero thresholds. Specifically, I apply thresholding to the connectivity signals in tasks where the number of positive examples is larger than the number of negative examples. The thresholds are selected for each task independently so that the numbers of positive and negative examples are approximately balanced. With this discretization of the connectivity signals, we are essentially predicting whether the connectivity strength between structures is above a certain threshold.

5.4 ENSEMBLE OF SPARSE AND TREE MODELS

I consider ensemble of sparse models for brain connectivity prediction. Given a set of training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ denotes the input feature vector, and $y_i \in \{-1, 1\}$ denotes the corresponding output label. In the brain connectivity prediction problem, $\mathbf{x}_i$ represents the structure-level gene expression vectors, and y_i encodes the structural connectivity information. The sparse models involve solving the following optimization problem [118, 131]:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} L(\mathbf{w}^T \mathbf{x}_i + b, y_i) + \lambda \|\mathbf{w}\|_1, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$ denote the model weight vector and bias term, respectively, and $\|\mathbf{w}\|_1$ denotes the vector ℓ_1 norm, defined as the sum of the absolute values of the elements, and λ is the regularization parameter.

It is well-known that the ℓ_1 norm regularization induces sparse model vector $\mathbf{w}$; namely some elements of $\mathbf{w}$ are set to zero. Thus, this regularization enables the integration of classification and feature selection into the same model. In the brain connectivity prediction problem, the features correspond to genes. Hence, the sparse formulation enables us to perform connectivity prediction and gene selection

simultaneously, thereby enabling the identification of most predictive genes for each brain structure. I employ the logistic regression loss function, defined as

$$L(\mathbf{w}^T\mathbf{x}_i + b, y_i) = \log\left(1 + \exp\left(-y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)\right), \tag{2}$$

as this loss has been shown to yield competitive performance in classification tasks [38, 87, 104]. The objective function in Eq. (1) is convex and can be solved by a variety of algorithms [130], including the proximal gradient method [89] and the alternating direction method of multipliers [17].

The parameter λ in Eq. (1) controls the amount of regularization. When λ is set to a small value, most elements of $\mathbf{w}$ will not be zero, giving rise to a full (non-sparse) model. On the other hand, when λ is set to a large value, many elements of $\mathbf{w}$ will be set to zero, leading to decreased predictive power. Hence, selecting an appropriate value for λ is important. I employ an ensemble learning technique known as stability selection for robust feature selection in sparse models [23, 96].

In stability selection, a set of λ values $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ are selected, thereby allowing models with different degrees of sparsity to be generated. To enable robust estimation, data sets of size $\lfloor n/2 \rfloor$ are repeatedly sampled, without replacement, from the original data of size n. For each sampled data set, a set of models, corresponding to different λ values in Λ, are obtained. Then I can compute the selection probability for the jth feature as

$$\Pi_j^{\lambda_i} = P(w_j \neq 0 | S, \lambda_i), \tag{3}$$

where S denotes the set of sampled data sets. The selection probability captures the relative frequency that each feature is selected among the models generated from sampled data sets under a particular regularization value λ_i. Based on the selection probability, a set of stable features can be identified as

$$F^{\text{stable}} = \left\{ j \,\middle|\, \max_{\lambda_i \in \Lambda} \Pi_j^{\lambda_i} \geq \pi_{\text{thr}} \right\}, \tag{4}$$

where $0 < \pi_{\text{thr}} < 1$ is a threshold value. It has been shown in [96] that, for a large range of π_{thr} values, the set of selected features are stable. In addition, a lot of theoretical guarantees and error control schemes have been presented, making stability selection an appealing method for feature selection in regularized models.

I also employ an ensemble of tree models, known as random forest (RF), for brain connectivity prediction. The purpose of use an additional method is to illustrate that different methods give similar predictive performance. A RF is a tree-based ensemble model for a variety of tasks, including classification, regression, and density estimation [7, 20, 37]. In the training phase, a set of randomized trees are built by introducing randomness in both the data and the tree construction processes. In the test phase, all trees are involved in making predictions, and the final decision is based on majority voting by all trees. This method has been applied in various biological [102], medical [36], and neurological [47, 52] applications. In applications such as brain connectivity prediction, there can be many features (genes), but only a small number of them contribute significantly to the prediction. An appealing property of RF is that it can produce an importance score for each feature that can be used for feature ranking and selection [6, 120].

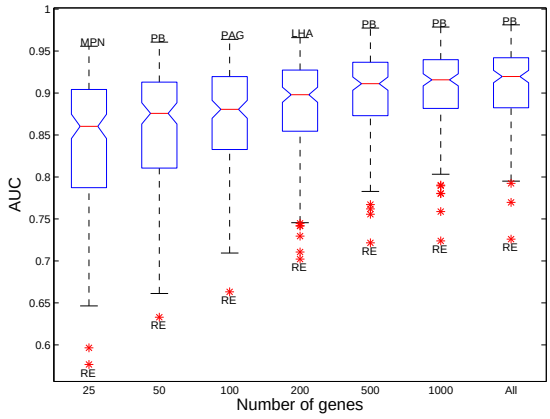
5.5 PERFORMANCE OF BRAIN CONNECTIVITY PREDICTION

In this section, I report the results of brain connectivity prediction using ensemble methods. For each prediction task, I apply five-fold cross validation and use the area under the ROC curve (AUC) as the performance measure [71, 127]. In this procedure, the samples are split into five (approximately) equally-sized subsets. Four subsets are used to train a model, and the fifth subset is used for performance evaluation. This process is iterated five times so that each of the five subsets is used in evaluation

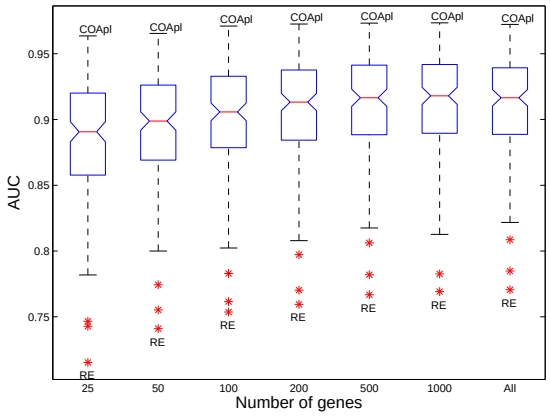
for exactly once. To identify genes that retain most of the predictive power for each structure, I rank the genes using both the tree and the sparse methods. I then use different numbers of top-ranked genes for making predictions. The ℓ_2 norm regularized logistic regression [86] is used in prediction based on sparse models.

I report the predictive performance obtained by the sparse and tree methods on the PID and AID data sets in Figure 13 using box plots. For each method, I report the performance using different numbers of top-ranked genes; namely 25, 50, 100, 200, 500, 1000, and all genes. It can be observed that, overall, I can achieve AUC values between 0.80 and 0.93, depending on the methods and the number of genes used. This indicates that gene expression information is predictive of connectivity in the mouse brain when the connectivity signals are discretized. I can also observe that a small subset of the 4084 genes are enough for giving high predictive power. I can observe from Figure 13 that the maximum performance difference between using all genes and 25 genes under different data sets and different methods is less than 5%, suggesting that a small number of genes are responsible for generating the structure-level brain connectivity. The performance difference between using all genes and 500 genes is less than 1%. my results are consistent with prior studies showing that a small number of genes give most of the predictive power [71, 127].

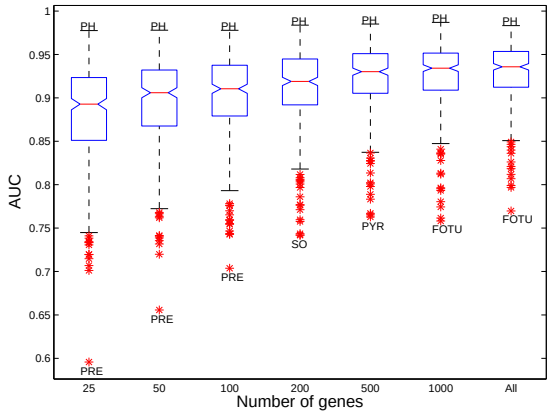
I observe from Figure 13 that, on average, sparse models and tree models give similar predictive performance. In addition, the performance differences between these two methods decrease monotonically as the number of used genes increases. When all genes are used, the differences on both data sets are less than 1%. This implies that the differences between tree and sparse models are mainly due to the differences in feature ranking. Empirical comparison between tree methods, logistic regression, and support vector machines [24] has been reported in the literature [28, 111]. The general conclusion is that their relative performance is application and data dependent.



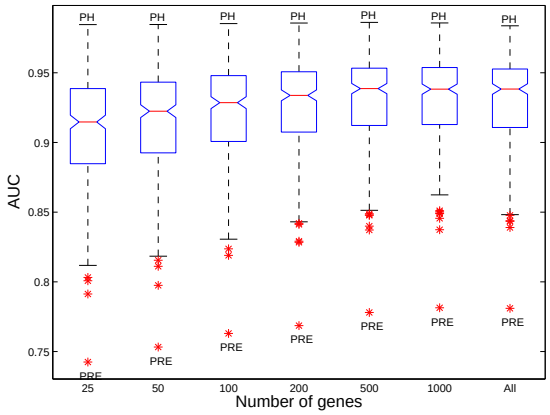
Sparse models, PID data



Tree models, PID data



Sparse models, AID data



Tree models, AID data

FIG. 13: Box plots of AUC values obtained by sparse and tree models on the PID and AID data using different numbers of genes. Each box represents the predictions on all structures using a particular method and number of genes. The central mark represents the median, the edges of the box denote the 25th and 75th percentiles. The whiskers extend to the minimum and maximum values not considered outliers, and outliers are plotted individually. In each case, the structures with maximum and minimum AUC values are annotated. Abbreviations: RE - Nucleus of reunions; MPN - Medial preoptic nucleus; PB - Parabrachial nucleus; PAG - Periaqueductal gray; LHA - Lateral hypothalamic area; COApl - Cortical amygdalar area, posterior part, lateral zone; PRE - Presubiculum; SO - Supraoptic nucleus; PYR - Pyramus (VIII); FOTU - Folium-tuber vermis (VII); PH - Posterior hypothalamic nucleus.

In addition, their performance difference is usually not very large. my results are consistent with those prior observations.

It has been shown in previous studies that spatial autocorrelation might exist between gene expression and connectivity [44, 45]. That is, spatially adjacent structures tend to be connected, and they might also have correlated expression patterns, since gene expressions are usually spatially localized. To elucidate the effect of spatial autocorrelation in connectivity prediction, I use baseline classifiers that determine whether structures are connected only based on their spatial proximity. That is, the proximities are used as classification scores in computing AUC. I use three proximity measures for this purpose; namely, average Euclidean distance, minimum Euclidean distance, and nomenclature distance. The average Euclidean distance measures the Euclidean distance between the centroids of brain structures [44]; the minimum Euclidean distance uses the minimum pairwise distance between all pairs of voxels in two structures as the structure-level distance; the nomenclature distance counts the number of shared ancestral structures in the ontology [44]. I report the prediction results using these three measures on the PID and AID data sets in Figure 14 along with that of my sparse models using all genes.

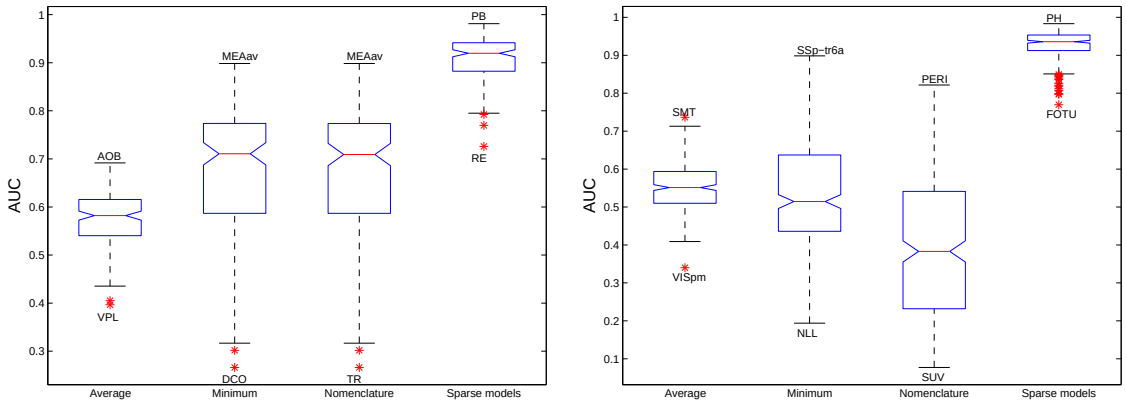
I can observe from Figure 14 that the sparse models give higher predictive performance than the baseline methods on both data sets. This indicates that the predictive values given by the sparse models are not the results of spatial autocorrelation. On the other hand, I can see that the baseline methods give better predictive values than that of random classifiers in five out of the six cases. This indicates that spatial autocorrelation exists between gene expression and connectivity. The classifier based on nomenclature distance gives the lowest performance on the AID data set. Recall that the AID data set is obtained by aggregating all sectional data sets containing a particular structure as either primary or secondary injection structures. Since a

sectional data set containing a structure as the secondary injection structure might contain only a small amount of injection in that structure but more injections in other structures, the aggregated AID data set might not accurately captures the connectivity of the designated structure. Thus, the nomenclature distance might not be an appropriate measure of structure-level distance on this data set. This might explain the low AUC value on the AID data set when the nomenclature distance is used.

5.6 COMPARISON WITH PRIOR STUDIES

Prediction of brain connectivity using gene expression data has been studied in the literature. Notably, [71] studied the connectivity prediction in the *C. elegans*, and they obtained AUC values of 0.594 and 0.601 in predicting the incoming and outgoing connectivity without gene selection. When gene selection is applied, they obtained AUC values of 0.60 and 0.61. [127] used gene expression data from the mouse brain to predict the connectivity in the rat brain. They achieved AUC values of 0.73 and 0.74 for incoming and outgoing connectivity prediction, respectively. I note that the methods and data used in [71, 127] and those in my work are substantially different. Thus, the results should not be compared in a direct, quantitative way. On the other hand, it is my experience that different prediction methods usually do not give very significantly different results. For example, [127] used support vector machines (SVM) for prediction and gene selection, and the performance by SVM and that by logistic regression is usually similar. Hence, the major differences in results might be due to the differences in data.

The connectivity data used in [127] were from the Brain Architecture Management System (BAMS) [15]. BAMS contains data about connections between brain



PID data AID data

FIG. 14: Box plots of AUC values obtained by using the proximities of structures as prediction scores on the PID and AID data sets. We use three distance metrics, namely, average Euclidean distance, minimum Euclidean distance, and nomenclature distance, to measure the distance between structures. The AUC value of sparse models using all genes are also shown for the purpose of comparison. Each box represents the predictions on all structures using a particular method. In each case, the structures with maximum and minimum AUC values are annotated. Abbreviations: AOB - Accessory olfactory bulb; VPL - Ventral posterolateral nucleus of the thalamus; MEAav - Medial amygdalar nucleus, anteroventral part; DCO - Dorsal cochlear nucleus; TR - Postpiriform transition area; RE - Nucleus of reunions; PB - Parabrachial nucleus; SMT - Submedial nucleus of the thalamus; VISpm - posteromedial visual area; SSp-tr6a - Primary somatosensory area, trunk, layer 6a; NLL - Nucleus of the lateral lemniscus; PERI - Perirhinal area; SUV - Superior vestibular nucleus; FOTU - Folium-tuber vermis (VII); PH - Posterior hypothalamic nucleus.

structures in the rat that were either collated from the literature, or inserted by neuroscientists based on their own experiments. The rat connectivity matrix in BAMS is very sparse, and only about 10% of the entries are filled [15]. The missing entries represent either no connection or lack of report, and both cases were considered as "no connection" in the prediction studies. In contrast, my data are from the same species, and my connectivity data are generated by systematic experiments and thus are expected to be more accurate and complete. In addition, I apply discretization to the connectivity signals, and the numbers of positive and negative samples in my prediction tasks are approximately balanced. This step might make the classification tasks easier and thus leads to high performance. The performance difference might also be contributed by the fact that [127] used data from two species, *i.e.*, mouse and rat, and the brain connectivity in rat and mouse might be different.

I also compare the top-ranked genes identified by [127] and those generated from the sparse model in my study. Specifically, [127] used data for 20936 genes from the sagittal section, while I use data for 4084 genes from the coronal section. [127] reported the top 500 genes for all brain structures, and only 81 of them are in the set of 4084 genes I use. Following [127], I also identify the top 500 genes for all structures using my sparse method and ABA data. I observe that, out of the 81 common genes, only 8 and 7 of them are common for the PID and AID data sets, respectively. This shows that the top-ranked genes given by prior and my studies are substantially different. This result is not surprising given the substantial differences in predictive performance. I expect that such differences are primarily due to the differences in the data used in these studies.

TABLE 9: Frequencies of gene ontology terms on the PID and AID data sets. Cellular component and biological process terms enriched in the top 200 genes identified by the sparse models are shown.

	Cellular Component		Biological Process	
PID	cell projection	67	multicellular organismal signaling	7
	neuron projection	58	synaptic transmission	5
	synapse	38	transmission of nerve impulse	4
	cell projection part	16	regulation of exocytosis	2
	dendrite	12	system process	2
	synapse part	11	neurological system process	2
	cytoplasmic part	9	regulation of calcium ion-dependent exocytosis	1
	cytoplasm	9	neurotransmitter secretion	1
	cell body	4	establishment of localization in cell	1
	dendritic spine	3	single-multicellular organism process	1
	neuron spine	3	multicellular organismal process	1
	perinuclear region of cytoplasm	3	cell-cell signaling	1
	axon	3	regulation of biological quality	1
	cytoskeleton	3	establishment of protein localization	1
	axon part	2	vesicle-mediated transport	1
AID	neuron projection	140	transmission of nerve impulse	13
	cell projection	139	multicellular organismal signaling	12
	synapse	91	synaptic transmission	8
	dendrite	43	neurological system process	3
	cell projection part	41	system process	3
	synapse part	38	single-organism reproductive behavior	1
	cytoplasmic part	27	organic substance transport	1
	cytoplasm	19	cell-cell signaling	1
	axon	8	vesicle-mediated transport	1
	dendritic spine	8	regulation of calcium ion-dependent exocytosis	1
	neuron spine	8	establishment of localization in cell	1
	cell junction	7	regulation of neuronal synaptic plasticity	1
	synaptic membrane	6	secretion by cell	1
	axon part	5	regulation of exocytosis	1
	cell body	5	regulation of transmission of nerve impulse	1

5.7 ENRICHMENT ANALYSIS OF HIGHLY RANKED GENES

An appealing property of the ensemble models is that they can identify genes that are highly predictive of brain connectivity. These highly ranked genes are expected to be involved in signal transmission processes and thus would function in the neuronal connectivity structures, such as the synapses, dendrites, and axons. To validate this hypothesis, I use the gene ontology cellular component and biological process domains [9] to analyze the highly ranked genes generated by the ensemble models. For each brain structure, I retrieve the highly ranked genes and use the hypergeometric distribution to compute enriched gene ontology terms [18]. I observe that the genes identified by the sparse models tend to yield more enriched terms than those generated by the tree models. This result is consistent with my observation in the predictive analysis, where the genes identified by the sparse models tend to give higher predictive performance. I thus focus on the ensemble of sparse models in the rest of this study.

I repeat this analysis for different numbers of highly ranked genes and find the results are largely consistent. When the top 200 genes of each structure identified by the sparse models are used in the analysis, 110 out of the 160 structures in the PID data set and 211 out of the 301 structures in the AID data set give at least one enriched cellular component term with corrected p-values less than 1%. I can see that many of the brain structures yield enriched terms. To provide an overview of these enriched terms, I collect the list of enriched terms for each brain structure in the PID and AID data sets. I then tabulate the frequency of each enriched term, and the ranked lists of the most frequent 15 terms are reported in Table 9. I can see that most of the frequently enriched cellular component terms refer to structures related to brain connectivity, including cell/neuron projection and junction, synapse,

TABLE 10: Marginal, conditional, and partial correlations between the gene expression and the connectivity at the structure level. The conditional correlations are computed by the graphical LASSO, and the partial correlations are the results of partial Mantel tests. We use three distance measures, namely average Euclidean distance, minimum Euclidean distance, and nomenclature distance, in the partial Mantel tests.

	PID	AID
Marginal correlation	0.3259 (p-value$< 10^{-3}$)	0.3583 (p-value$< 10^{-3}$)
Graphical LASSO	0.6178 (p-value$< 10^{-3}$)	0.6107 (p-value$< 10^{-3}$)
Average Euclidean distance	0.3258 (p-value$< 10^{-3}$)	0.3582 (p-value$< 10^{-3}$)
Minimum Euclidean distance	0.1513 (p-value$< 10^{-3}$)	0.1777 (p-value$< 10^{-3}$)
Nomenclature distance	0.1961 (p-value$< 10^{-3}$)	0.2087 (p-value$< 10^{-3}$)

dendrite, and axon. This shows that the products of many highly ranked genes are localized to structures related to brain connectivity and thus may be involved in related processes. Indeed, I can also see that most of the enriched biological process terms are related to signaling, synaptic transmission, nerve impulse transmission, neurotransmitter secretion, and molecular transport. These results indicate that the highly predictive genes are commonly localized in neuronal projections and involved in biological processes related to inter-neuron communication.

5.8 CORRELATION BETWEEN GENE EXPRESSION AND CONNECTIVITY

My results show that gene expressions in the mouse brain are highly predictive

of brain connectivity. It has also been shown that gene expressions in the mouse brain are correlated to connectivity in the rat brain [44, 45]. It is thus illuminating to study the correlation between gene expression and connectivity in the mouse brain using my systematic data. To this end, I compute two correlation matrices that capture the pairwise correlations between structure-level gene expression patterns and connectivity patterns, respectively. I then compute the correlations of these two matrices and perform Mantel test to assess the statistical significance of the correlation between gene expression and connectivity. This result represents the marginal correlation between gene expression and connectivity.

One factor that needs to be considered in this analysis is the effect of spatial autocorrelation. The marginal correlation is the pairwise correlation coefficients, and thus the results might be attributable to spatial autocorrelation. To address this issue, we compute the conditional and partial correlations between gene expression and connectivity patterns. The conditional correlations are computed by applying the penalized Gaussian graphical models known as graphical LASSO [10, 46], and thus the effect of spatial autocorrelation is circumvented. The partial correlations are computed by applying partial Mantel test [83, 108] as was done in [44]. I employ the three distance measures used in the predictive analysis in partial Mantel test. The p-values are computed by randomizing the data and repeating the experiments 1000 times [44]. The results on both the PID and the AID data sets are reported in Table 10. I can observe that the marginal, conditional, and partial correlations are all significant. These results show that the gene expressions and connectivity are significantly correlated in the mouse brain.

TABLE 11: Correlations between the number of shared, highly-ranked genes and the spatial distance at the structure level. The p-values are the results of Mantel tests.

# of genes	PID		AID	
	Correlation	p-value	Correlation	p-value
25	0.0700	$<10^{-3}$	0.0415	0.001
50	0.0670	0.003	0.0647	$<10^{-3}$
100	0.0846	$<10^{-3}$	0.0833	$<10^{-3}$
200	0.0989	$<10^{-3}$	0.1013	$<10^{-3}$
500	0.0886	0.001	0.0791	$<10^{-3}$
1000	0.0372	0.106	0.0261	0.073

5.9 STRUCTURE-LEVEL CORRELATION OF HIGHLY-RANKED GENES

In this study, I predict the connectivity of each brain structure separately. Hence, I obtain a list of ranked genes for each structure by applying the ensemble models. An interesting question would be to study the overlapping of highly-ranked genes between structures and correlate the results with other information. To perform this analysis, I compute the number of common genes in the ranked lists between pairwise structures and encode the results into a matrix. I study how this matrix is correlated with the matrix encoding spatial distances of structures. To this end, I employ the minimum Euclidean distance measure and perform Mantel tests to assess the statistical significance of the correlation between these two matrices. The correlations and p-values for the sparse models with different numbers of genes are

reported in Table 11. I can observe that most of the p-values indicate significant results. This shows that the highly-ranked genes tend to overlap for spatially adjacent brain structures.

5.10 COMPARISON BETWEEN PID AND AID DATA SETS

I observe from Figure 13 that, with the same method and the same number of genes, the performance on the AID data set is consistently higher than that on the PID data set. Recall that the AID data set was obtained by aggregating all grid data that include a particular brain structure as either primary or secondary injection structures. In contrast, the PID data set was obtained by only considering the primary injection structures. Hence, the AID data are the results of aggregating more sectional data sets and thus might contain more complete information. On the other hand, since majority of the injections are restricted to the primary injection structure, the difference between these two data sets is small.

CHAPTER 6

HIGH-RESOLUTION PREDICTION OF CONNECTOME USING GENOME

In this chapter, I study the relationship between transcriptome and connectome in the adult mouse brain at a high-resolution voxel-level. Previous studies on the mammalian brain were either performed on data coming from different species due to insufficient resources or were performed at the coarse regional level. Since the brain regions are defined as a hierarchy with very diverse sizes, measuring and analyzing gene expression and connectivity at this level might lead to less quantitative analysis and inaccurate results. It would be interesting to perform analysis at a fine voxel level as this would avoid any overlapping between nested brain regions in the hierarchical ontology, and the results are less sensitive to region sizes.

My high-resolution analysis is made possible by the recent release of voxel-level gene expression and brain connectivity data from the Allen Brain Atlas (ABA) [113]. The ABA provides gene expression signatures for approximately 4,000 coronal genes across all brain voxels with a very high resolution [3]. I incorporate those gene expression data with the high resolution connectivity data obtained from the Allen Mouse Brain Connectivity Atlas [2] to set the foundation for integrative analysis in a single mammalian brain. I use sparse modeling to predict connectivity signatures based on the gene expressions at the voxel-level. The use of sparse modeling enables me to perform prediction and feature selection simultaneously, leading to lists of genes that are mostly responsible for the connectivity prediction.

6.1 BACKGROUND AND MOTIVATION

The mammalian brain contains a wide variety of cells, connected into an interaction network that collectively generate an organism's cognitive functions and determine its behavior. Understanding the functional circuitry of this enormously complex machinery plays a central role in modern neuroscience. An increasing amount of evidence indicates that correlations in gene expression patterns are predictive of structural connectivity, prompting the systematic cataloguing and analysis of expression and connectivity patterns.

The initial studies investigating the relationship between transcriptome and connectome primarily focused on the worm *Caenorhabditis elegans* [11, 71, 123]. The *C. elegans* has a simple nervous system, and its synaptic connectivity has been fully identified. [71] found that gene expression of a neuron carries significant information about its synaptic connectivity. In addition, they identified a list of putative genes that might be able to predict connectivity. [123] have conducted their experiments on *C. elegans* to identify sets of genes whose joint expression is common to most synapses. In [106], several interesting correlations between the gene expression and connectivity in the *C. elegans* have been identified.

The interesting results obtained from *C. elegans* prompted similar analysis on more complex species. [127] carried out their experiments on the rat and mouse brains due to the lack of data from the same species. They were able to predict the regional connectivity based on the gene expression signatures for many brain regions with significant performance. They also identified lists of genes that significantly contribute to connectivity prediction and compared them to the genes known from literature to be related to different brain diseases. [44, 45] used a series of covariation analysis techniques to demonstrate that gene expression in the mouse brain is correlated to

connectivity in the rat brain. Also, I have demonstrated that gene expressions in the mouse brain are highly predictive of brain connectivity at the coarse region level in 5.

My results in this chapter indicate that gene expression and connectivity signature are highly correlated at the voxel level. I can predict connectivity based on transcriptome with an accuracy of 93%. Very high prediction accuracy can even be achieved using a very small number of genes. I perform feature selection to identify the top genes responsible for the high prediction accuracy. I analyze the biological process and cellular component term enrichment of those top genes to provide a deeper understanding of their functionality. The term enrichment analysis confirms that those genes are involved in different connectivity-related processes. I further show that the highly predictively genes are enriched more in neurons than in glia. To provide in-depth analysis of the transcriptome-connectome relationship, I study the correlation between gene expression patterns and connectivity target specificity. My results show that gene expressions and connectivity target specificity are significantly correlated after spatial autocorrelation is considered.

6.2 EXPERIMENTAL DESIGN

I am using the data provided by the Allen Brain Atlas (ABA) [113], which contains gene expression and connectivity data for the adult mouse brain. The gene expression data is available through the Allen Mouse Brain Atlas [84], and the connectivity data is provided in the Allen Mouse Brain Connectivity Atlas [2]. Sample images from the connectivity atlas are shown in Figure 15.

The ABA provides an annotated 3D reference model. This annotated 3D space is the one upon which all the ISH and projection sectional data sets are aligned. The reference atlas enables an integrated analysis of different data modalities like gene

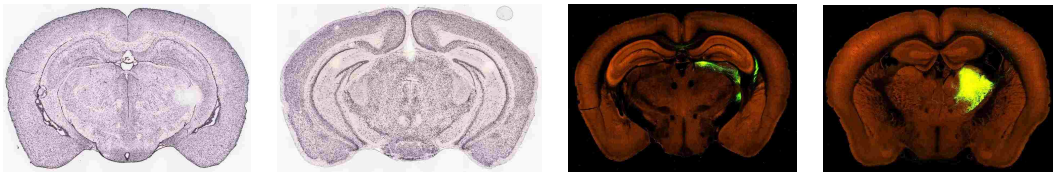


FIG. 15: Sample images from the Allen Mouse Brain Atlas (left two) and the Allen Mouse Connectivity Atlas (right two) data sets. The ISH images correspond to *Apoe* and *Nap1l5* coronal genes respectively. The projection images correspond to injections in the *Posteromedial visual area* and *Reticular nucleus of the thalamus* in the adult mouse brain respectively.

expression and projection images. The informatics data processing pipeline then converts the ISH and projection images to grid-level voxel data at resolutions of 100 μm and 200 μm that is used throughout my experiments.

The goal of this work is to study the gene expression and connectivity of the mouse brain simultaneously at the voxel level. In my experiments, I extract gene expression signatures from the Gene Expression Atlas and projection data from the Connectivity Atlas. The gene expression signatures represent the expression energy values of genes at the 200 μm grid resolution level. The 200 μm level annotation contains approximately 60,000 annotated voxels that form the rows of the gene signature matrix. The Gene Expression Atlas contains sectional data sets for approximately 4,000 genes. The energy values in each sectional data set corresponding to a specific gene are extracted to form the columns of the gene signature matrix. Some genes might have multiple sectional data sets representing multiple experiments for the same gene. In this case, I chose to take the average of the energy values across all

section data sets for the same gene [127].

Similar to the Gene Expression Atlas, the projection energy values of the annotated voxels are extracted from the projection sectional data sets corresponding to injections in different structures. The connectivity data sets are processed at 100 μm grid resolution, and the volumetric data is downsampled to 200 μm grid resolution in order to match the resolution of the gene expression data. Each sectional data set forms a column of the connectivity matrix and is considered as a label vector for a prediction task. Each entry in the label vector specifies either connection (positive sample) or disconnection (negative sample) from the injection site to the corresponding voxel. My goal is to predict this label vector based on the gene expression matrix. This formulation is illustrated in Figure 16.

The Connectivity Atlas provides 1019 sectional data sets representing injections from 209 unique primary injection structures. I average the sectional data sets with the same injection structure to obtain 209 connectivity data sets. I found that 19 out of the 209 data sets either have nonzero connectivity energy values to more than 90% or less than 10% of the rest of the brain voxels. If those regions were to be included in our experiments, I would have had an excessively unbalanced number of positive and negative samples in classification tasks. I chose to exclude those data sets, resulting in 190 prediction tasks. We consider any nonzero projection energy value as an indication of connection (positive sample) and any zero projection energy value as absence of connection (negative sample).

The prediction is performed by building a computational model on some of the samples. This model is then applied to the remaining samples to generate connection probabilities, which are then compared with the true label vector to test the prediction accuracy. To remove false positive samples due to the injections, I remove all the injection voxels corresponding to this task from my feature and label matrices. These

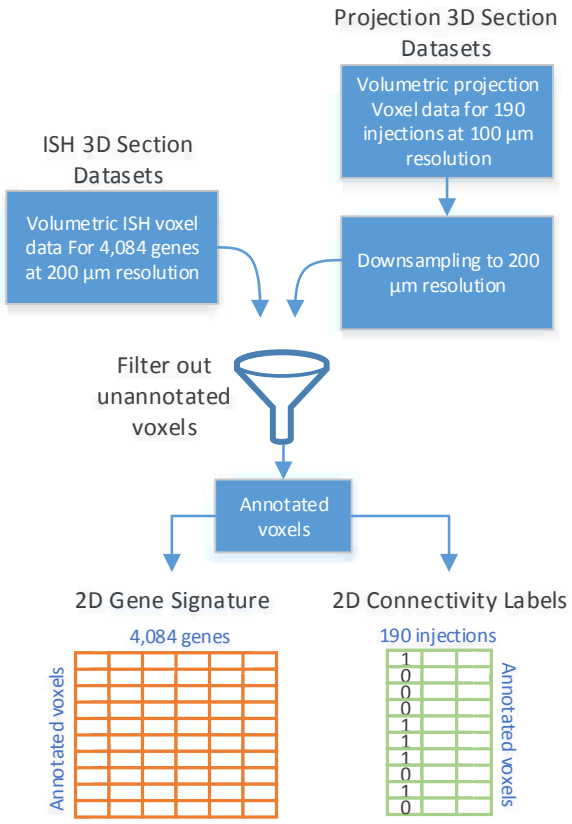


FIG. 16: Illustration of the connectivity prediction pipeline.

voxels might reflect positive labels due to injection rather than connectivity.

6.3 SPARSITY METHOD

I use sparse linear models for connectivity prediction; as such models yield competitive performance while retaining interpretability. In order to identify the most

predictive genes while making accurate predictions, I employ a sparse linear model in connectivity modeling. The ℓ_1-norm regularized models, known as LASSO [118], are a class of powerful methods for learning interpretable models. These sparse models set some of the model parameters to zero as a result of minimizing the ℓ_1-norm of the parameter vector. Since each element in the parameter vector corresponds to a specific gene, setting an element of the parameter vector to zero is equivalent to eliminating the corresponding gene from my model.

In the experiments, I tune the classifier to keep the number of selected genes in my model to be between 50 and 250. One remarkable observation is that when I re-applied linear models on my data using only the selected genes from the sparse model, I obtain almost the same accuracy as was obtained from using all the genes. This served as a way to determine the most significant genes identified by my feature selection model.

6.4 PREDICTION AND GENE SELECTION

In this section, I provide the results of the connectivity prediction experiments. In addition, I perform feature selection using sparse models to obtain the top genes that contribute most to the prediction performance. The prediction is then recalculated based on only the top genes selected. I demonstrate the injection sites correlation in terms of the gene signature and connectivity after removing the spatial autocorrelation effect. I also analyze the selected genes in terms of their Gene Ontology and cell-type enrichments.

I use an evaluation approach in which 2/3 of the samples are randomly selected and used to construct the model. Those randomly selected samples are the training samples for a specific task and are never used simultaneously as a testing sample for the same

TABLE 12: Average AUC and standard deviation calculated using the sparse models for different sets of genes selected by stability selection. The results by baseline classifiers are also reported.

Gene count	25	50	100	200	500	All	Min. BL	Eucl. BL
Avg. AUC	0.8369	0.8677	0.8847	0.8953	0.9086	0.9380	0.5698	0.79017
Std. deviation	0.0448	0.0345	0.0303	0.0286	0.0261	0.0199	0.0889	0.0616

task. Multiple sparse models are constructed from the training samples by randomly choosing a sub-sample of size $\lfloor N/2 \rfloor$ in each randomization and for different λ values ranging between 0.0001 and 1 to induce different levels of sparsity. The remaining 1/3 of the samples (testing samples) were used to test the predictive accuracy. The accuracy is measured by the area under the receiver operating characteristic (ROC) curve (AUC). The ROC plots the fraction of true positives versus the fraction of false positives for a binary classifier system, while its discrimination threshold varies.

I use the training data to perform stability selection and gene ranking. The AUC values obtained from using the top 25, 50, 100, 200, 500 and all 4084 genes are reported in Table 12. I also study the statistical significance of my results by calculating the p-values corresponding to using the top 25 genes for each injection structure and they were all less than 10^{-3}. I observed from the results that the predictive accuracy ranges between 83% and 93%. These very high accuracies show that gene expression signatures carry very significant information about the synaptic connectivity. Moreover, using a very small number of genes, I can obtain a very high predictive accuracy. This small number of genes carry the most information about

connectivity. These findings are consistent with prior results, which state that a small number of genes are responsible for most of the predictive accuracy [71, 127].

It has been reported in prior studies that spatial autocorrelation might explain brain connectivity to some extent [44]. That is, spatially adjacent regions tend to be connected, and connectivity prediction only based on proximity might give significant results. To investigate how much of my predictive performance is due to spatial autocorrelation, I construct two baseline classifiers. The first predicts connection probability only based on spatial proximity (Min. baseline classifier). In particular, this baseline classifier is built based on the minimum distance between each test voxel and all voxels in the injection site. Euclidean distance between every test voxel and all voxels in a specific injection site are computed and the minimum is set to be the distance between this voxel and the injection site. The other classifier is based on constructing a feature vector representing the Euclidean distance between the sample voxel and several reference voxels. It can be seen from Table 12 that the average AUC obtained from the minimum and the Euclidean baseline classifiers are 0.5698 and 0.79017 respectively. These numbers indicate clearly that a portion of the prediction performance is due to spatial autocorrelation. However, it is still less than the results reported by any sparse model for even a very small number of genes. This shows that the high prediction accuracy does not follow entirely from spatial autocorrelation but is derived from the gene expression patterns instead. A region by region AUC comparison of baseline classifiers with sparse models with different numbers of top genes is shown in Figure 17.

6.5 GENE ONTOLOGY TERM ENRICHMENT

Using my sparse models and stability selection, I am able to identify the genes

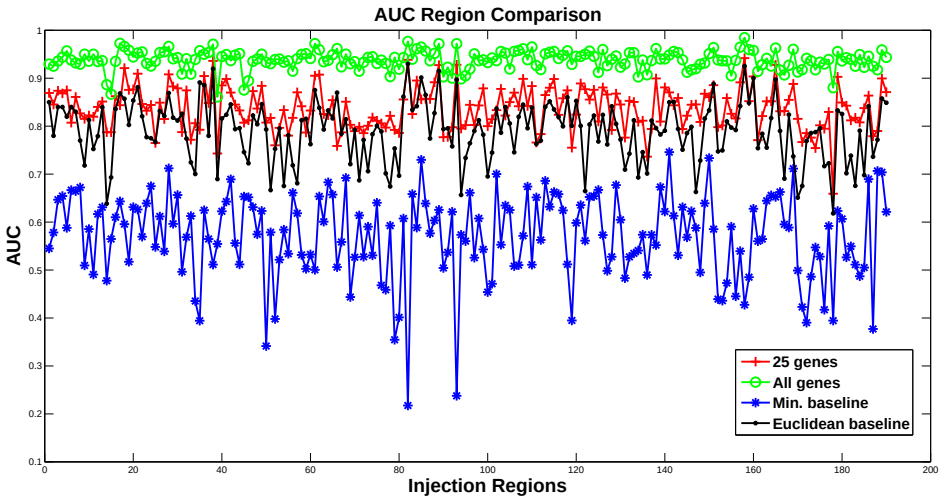


FIG. 17: The AUC values for every injection site. These results are achieved by using the top 25 genes, all genes along with those of the baseline classifiers based on minimum spatial proximity and Euclidean distance.

with high contribution to the predictive performance for each injection structure. I expect that these genes might carry critical information about synaptic connectivity and axon guidance. To provide an in-depth investigation in this direction, I analyze the biological process and cellular component term enrichment of those genes using the Gene Ontology (GO) [9].

I select the top 200 genes identified for each injection structure by stability selection and apply the GO analysis on them [18]. I identify all terms with corrected p-values less than 1%. I count the frequency of each term across all injection structures, and I observe that most of the biological process terms are related to transport, multicellular organismal signaling, establishment of localization and synaptic transmission. In

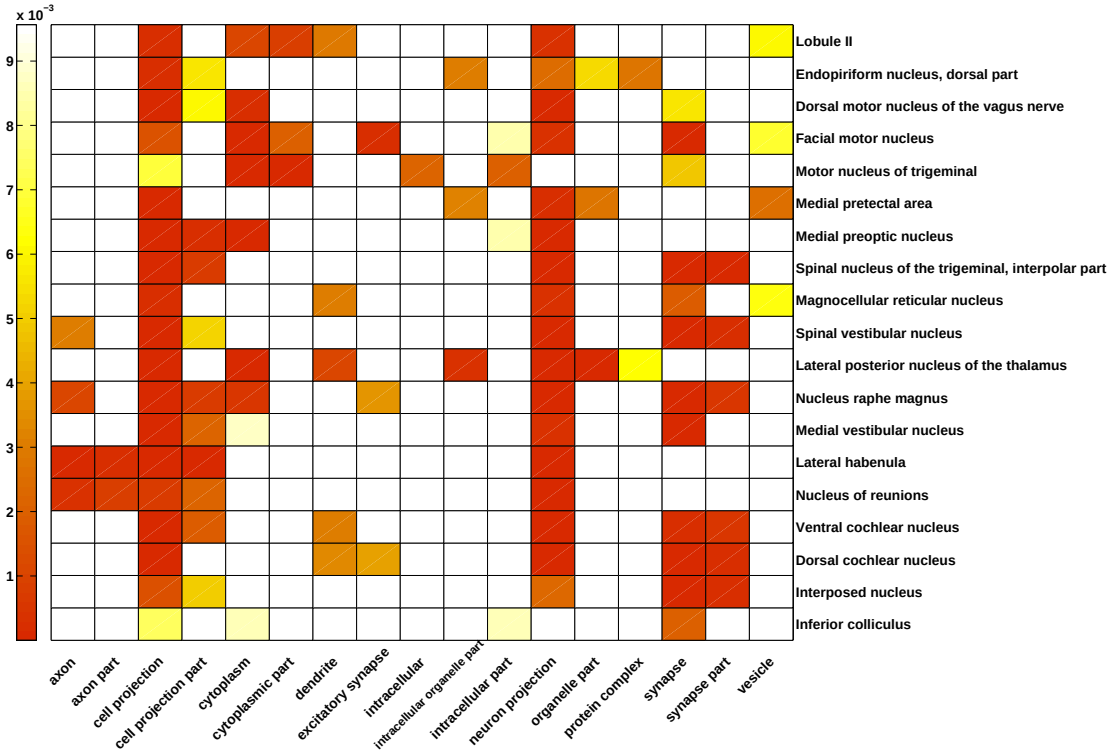


FIG. 18: Gene Ontology cellular component term enrichment p-values for different injection structures.

addition, the cellular component terms are mostly associated with cell projection, neuron projection, synapse, dendrite and axons.

The GO enrichment results are presented for only the top 200 genes for the sake of compatibility with prior studies. I also assess the GO enrichment experiments on different number of genes ranging between 25 and 500.

To provide a snapshot of the most frequent terms and the associated injection structures, I depict the term-structure associations in Figures 18 and 19. For the

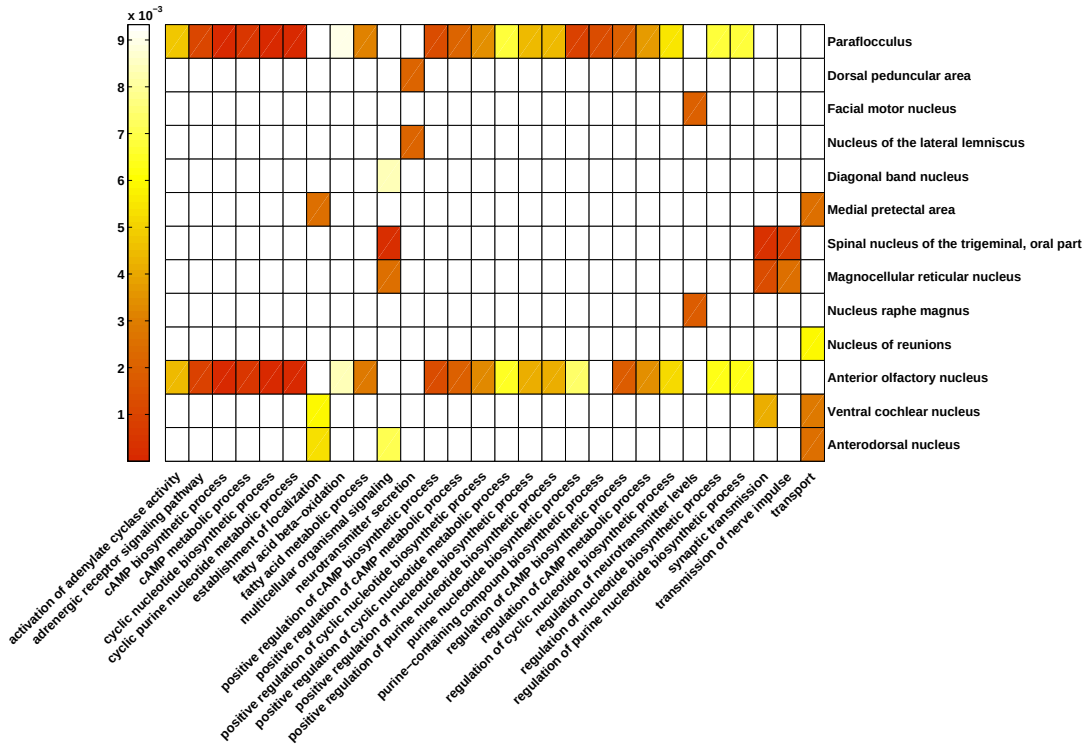


FIG. 19: Gene Ontology biological process term enrichment p-values for different injection structures.

biological process terms, I only show the terms that are enriched in at least two injection structures while I display all the injection structures having at least 1 enriched term. On the other hand, because the cellular component analysis gives a lot more terms than the biological process analysis, I display only the terms enriched in at least 4 injection structures and injection structures that have at least 5 enriched terms. I can see that these top genes selected by the stability selection play important roles in cell and neuron projections as well as axon guidance and inter-neuron communication.

6.6 GENE EXPRESSION AND CONNECTIVITY TARGET CORRELATION

My experiments show that gene expressions at connectivity target sites are highly predictive of connectivity from a particular injection site at the voxel level. Recent experimental results show that gene expression patterns at the injection sites are correlated with the connectivity target specificity in the primary somatosensory cortex [109]. To explore brain-wide correlation analysis, I construct two correlation matrices; the first one captures the correlations between gene expressions at the injection sites and the second one captures the correlation of connectivity from the corresponding injection sites to other brain voxels. One important factor that should be accounted for is the distance between those injection sites and its effect on this experiment. To eliminate the spatial autocorrelation effect, I construct a third matrix that represents the pairwise distance between injection sites. I perform a partial Mantel test [83] to measure the correlation significance of the two correlation matrices while eliminating their spatial autocorrelation.

Specifically, the pairwise gene correlation of two injection sites is calculated by first extracting the gene signature of all injection voxels belonging to each injection site. This is done by down-sampling the injection voxels of each injection site to the 200 μm level resolution to match the resolution of the gene expression data. Each injection site is then represented by a gene expression signature vector obtained by averaging the gene signature of its own injection voxels. The correlation between those two vectors is then computed to capture the gene expression correlation between those two injection sites. On the other hand, the connectivity correlation is calculated by first removing the injection voxels of the two sites from both projection sectional data sets. The correlation between the remaining annotated voxels in both sectional data sets then represents the connectivity target correlation of those two sites. To

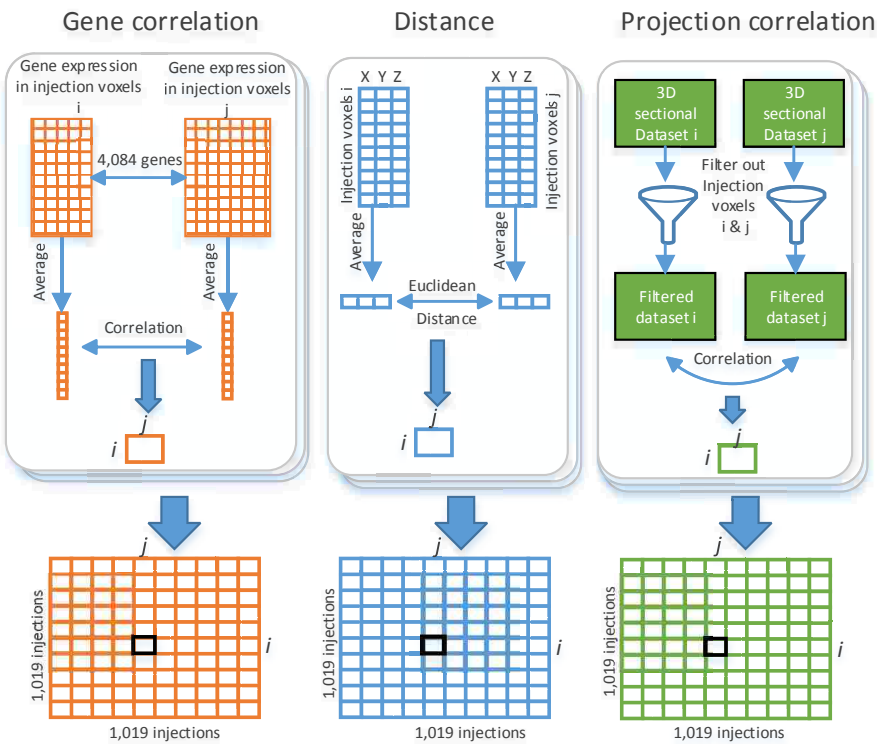


FIG. 20: Illustration of the steps for obtaining the three matrices used in the gene expression and connectivity target specificity correlation study.

calculate the injection site distance, I use the average Euclidean distance between the voxels belonging to each of the two injection sites to represent their pairwise distance. The steps for obtaining these matrices are illustrated in Figure 20.

The partial Mantel test yields a correlation of 0.2278 with a significant p-value of 0.001. In detail, the p-value is calculated by randomizing the data 1000 times and calculating the probability that the same or higher correlation value is achieved by the randomized data. My result shows that the gene expression patterns at injection

TABLE 13: Cell-type enrichment percentages for the top 25, 50, 100, 200, and 500 genes identified in my study. The reported numbers represent the percentage of the top genes identified in my study that overlap with the 3 lists of genes enriched in astroides, neurons, and oligodendrocytes along with their corresponding p-values. The percentages are the average value across all injection structures. Only enrichment results for neurons appear to be significant while for the other two cell types i, the enrichment is random

# of top genes	Astroides	Neurons	Oligodendrocytes
25	17.79, p-value=0.46	39.07, p-value=0.008	9.07, p-value=0.653
50	14.74, p-value=0.652	36.09, p-value=0.002	11.16, p-value=0.668
100	14.57, p-value=0.761	36.09, p-value$< 10^{-3}$	12.08, p-value=0.585
200	14.93, p-value=0.854	36.01, p-value$< 10^{-3}$	12.66, p-value=0.57
500	15.38, p-value=0.91	34.58, p-value$< 10^{-3}$	13.25, p-value=0.451

sites and the corresponding connectivity target are highly correlated. This study extends the results in [109] to a brain-wide analysis.

6.7 CELL-TYPE-SPECIFIC GENE ENRICHMENT

The brain consists of two types of cells; namely neurons and glia. It is believed that neurons are responsible for intercellular communication and connection while glial cells provide auxiliary support functions to neurons. I thus expect that the top genes identified in my study are mainly expressed in neurons, since they are responsible for generating brain connectivity. To validate this hypothesis, I use the

TABLE 14: Correlations and p-values between the common gene matrix and the spatial distance matrix. The p-values are the results of Mantel tests. The common gene matrix is computed based on the number of common genes in the top 25, 50, 100, 200 and 500 genes selected per injection structure.

# of top genes	Correlations	p-values
25	0.2007	$< 10^{-3}$
50	0.2553	$< 10^{-3}$
100	0.2885	$< 10^{-3}$
200	0.2517	$< 10^{-3}$
500	0.1340	0.003

cell-type-specific genes for neurons along with other cell types like astrocytes and oligodendrocytes identified in [25]. After removing genes that are not in the ABA coronal gene set, I obtain 711 genes enriched in astrocytes, 776 genes enriched in neurons, and 541 genes enriched in oligodendrocytes.

I compare the overlap of the top genes identified in my study with these cell-type-specific genes for each injection structure. In particular, I calculate the number of common genes between my top genes and each of the 3 lists of genes corresponding to different cell types for each injection structure. The average enrichment percentage for each cell type across all injection structures is then calculated. To ensure that those numbers are statistically significant, I calculated the p-values by randomly selecting a set of genes and calculating their enrichment percentage per each cell type for different number of genes. While the average enrichment percentage for neurons

is constantly higher than that for other cell types, only the neurons enrichment percentages were statistically significant. I show the average enrichment percentage across all injection structures for different gene list sizes in Table 13. The results indicate that, regardless of the number of top genes included, the top genes identified in my study are consistently enriched in neurons with a high statistical significance. Also, these genes do not show any enrichment significance with other cell types. These findings show that the most discriminative genes identified by my feature selection technique are mostly expressed in neurons and not necessarily expressed in other cell types like astrocytes and oligodendrocytes. I also observe that, as the number of top genes increases, the enrichment percentages for neurons decrease.This shows that the very top genes are more highly enriched in neurons.

6.8 STRUCTURE-GENE CORRELATION

I next study the correlation between the top genes and spatial distance of the injection structures, since I expect that spatially adjacent structures would share their top genes. To this end, I calculate the common gene matrix that captures the percentage of common genes between each pair of injection structures. I also compute the spatial distance matrix as the minimum Euclidean distance between each pair of voxels in the respective injection structures. I perform the Mantel test to assess the correlation between the common gene matrix and the spatial distance matrix, and the results are reported in Table 14.

It is clear from the results that there are significant correlations between the spatial distance matrix and the common gene matrix. Specifically, for different numbers of

top genes considered in the common gene matrix calculation, the correlations are consistently significant. This shows that the highly ranked genes tend to overlap for spatially adjacent brain structures.

CHAPTER 7

GLOBAL ANALYSIS OF GENE EXPRESSION AND PROJECTION TARGET CORRELATIONS

In this chapter, I conducted a global, quantitative analysis of gene expression and projection target correlations in the adult mouse brain. I mainly focused on studying how the gene expression patterns in the source neurons are globally related to projection target specificity. In this sense, my study is fundamentally different from the prior ones reported in [44, 45, 127]. Instead, my work was mainly motivated by [109] and aimed at a global, quantitative analysis that is lacking to date.

7.1 BACKGROUND AND MOTIVATION

The functions of neurons are largely determined by their molecular compositions. Those molecules are encoded by the genome that is expressed uniquely in each neuron. The mammalian brain contains a large number of neurons that are connected in diverse patterns, resulting in complex interaction networks that control information flow. In those interaction networks, neurons typically have very diverse projection target specificity. For example, projections from the primary somatosensory cortex (S1) target both cortical and subcortical regions [109]. Also, it has been shown that cortico-cortical projections in the mouse visual cortex are also functionally target specific [50]. To obtain a better understanding of the diversity of projection neuron classes, transcriptome analysis of the neurons along with a direct correlation with projection targets are needed.

The integrative analysis of neuronal gene expression and connectivity patterns was initially carried on the worm *Caenorhabditis elegans* as its gene expression and neuron-level connectivity are simple and largely known [11, 71, 123, 126]. Those studies showed that the genetic properties of neurons significantly influence their synaptic network structures. [71] performed a co-variation correlation experiment known as Mantel test and illustrated that gene expression and connectivity patterns are significantly correlated. A similar analysis was performed later on the mammalian brain, leading to more significant results [41, 44, 45]. Specifically, [44] carried out a large-scale analysis of the transcriptome-connectome correlation in the rodent brain, leading to a correlation of 0.25. These high correlations inspired other studies to even predict the connectome based on the gene expression patterns. [127] performed this prediction with an accuracy up to 83% in the rodent brain. In addition, they identified many genes that contribute most to this high prediction. Such analysis has recently been extended to the human brain [51].

The abovementioned studies focused on analyzing how the gene expression patterns of source and target neurons are correlated as compared to neurons that are not connected. The prediction studies used the expression patterns of target neurons to predict their connectivity with a particular source neuron. On the other hand, increasing evidence has shown that there are also direct correlations between source neuron gene expression patterns and projection target specificity [76]. In a recent study, efforts have been made to identify genes that are expressed in specific excitatory projection neuron classes [109]. The study showed that the neocortex contains diverse populations of excitatory neurons that are definable by their specific cortical and subcortical projection targets. However, some of the most broadly used markers for specific layers were found not to be expressed selectively in neurons with a specific projection target. This indicates that in spite of the significant correlations between

marker genes and projection targets, the excitatory neuron projection targets are in fact diverse and complex [109].

By using the Allen Mouse Brain Atlas and the Allen Mouse Brain Connectivity Atlas data, I start this chapter by visualizing and clustering the injection site gene expression patterns and projection targets separately. These initial analysis showed that both data sets exhibit strong spatial autocorrelation. That is, nearby injection sites tend to express similar sets of genes and also tend to project to similar targets.

To account for spatial autocorrelation, I performed the partial Mantel test [83] in which the spatial effect is corrected. I found that even after correcting for the spatial autocorrelation, the two data sets are highly correlated with a partial correlation of 0.19. I adopted two greedy gene ranking approaches to identify the top genes responsible for this correlation. Using only the top genes identified by my gene ranking techniques in the correlation analysis, I was able to obtain a series of significant correlations with values up to 0.49. These results indicate that the voxel gene expressions directly affect their target projections. These results are consistent with the findings reported in [109], but have extended the previous study to a global and quantitative analysis.

7.2 EXPERIMENTAL DESIGN

In my experiments, I used two data sets from the Allen Brain Atlas (ABA) [113]. Specifically, I used data from the Allen Mouse Brain Atlas [84] and the Allen Mouse Brain Connectivity Atlas [100], which provide gene expression data and connectivity data, respectively, in the adult mouse brain. To allow an integrated study of both data sets, the ABA provides an annotated 3D reference model upon which all images from both atlases were aligned. Both atlases provide grid-level voxel data obtained

from images mapped to the same 3D reference space.

In the Allen Mouse Brain Connectivity Atlas (the Connectivity Atlas), axonal projections in the mouse brain are visualized by viral tracers from more than 200 regions. This atlas provides axonal projections along with injection voxel coordinates for 1,788 injection sites. I treated each injection data set independently throughout the experiments though some of the brain regions were injected multiple times, since the specific injection voxels are unique. In this atlas, the grid-level voxel data are provided at 100 μm resolution.

To perform an integrative analysis of gene expression patterns and projection targets, the gene expression and connectivity data sets should be mapped to the same space as they are originally provided in different resolutions. The data extraction and processing steps are illustrated in Figure 21. Specifically, the coronal gene expression data are provided for approximately 4,000 genes in a 3D grid-level format at a 200 μm resolution. For each gene, I extracted the energy values at the 60,452 voxels annotated in the reference atlas. The extracted voxels for each gene form a column of the gene expression data matrix. The connectivity data are provided for 1,788 injection sites at a 100 μm resolution. Similar to the gene data, I extracted the energy values at more than 400,000 annotated voxels from each projection data set corresponding to a specific injection site. Those extracted voxels form the columns of the projection data matrix. The two processed data sets were used later in our experiments to generate the injection sites gene correlation and projection correlation matrices.

To make an integrative analysis of the two data sets possible, the gene signature of each injection site is needed. I obtained the gene signature of each injection site by first down-sampling its injection voxels to the 200 μm resolution and then extracting the rows corresponding to those voxels from the gene signature matrix. The number of

injection voxels is usually different for different injection sites. I computed the average gene signature across all injection voxels to come up with a vector of approximately 4,000 genes representing the gene signature of a single injection site. This vector forms a column in the injection site gene signature matrix that was used later throughout my experiments. I observed that the energy values of the injection voxels are usually very high, as they represent injection values instead of projection energy. To eliminate this data artifact, I set the values of injection voxels to zero for each injection site independently.

7.3 DATA VISUALIZATION USING T-SNE

I intended to study the relationship between gene expression patterns and projection target specificity for different injection sites. To this end, I visualized the high-dimensional gene expression and projection target signatures associated with each injection site using the t-distributed stochastic neighbor embedding (t-SNE) method [94, 122]. t-SNE is an extension of SNE [58] to simplify the optimization and overcome the so-called 'crowding problem'. t-SNE aims to model local structures of high-dimensional data points while ensuring that global dissimilarity between clusters is preserved. To this end, t-SNE computes two similarity matrices; one is obtained based on symmetrized Gaussian conditional distributions of original data space, and one is computed from Student t-distributions of low dimensional space. The low dimensional data, known as map points, is learned by minimizing the Kullback-Leibler (KL) divergence between the probability distributions in the original data space and the embedding space. Since KL divergence is not symmetric, different types of mismatches contribute differently to the overall cost. As a result, nearby map points are produced to represent nearby original data points, while distant map points are

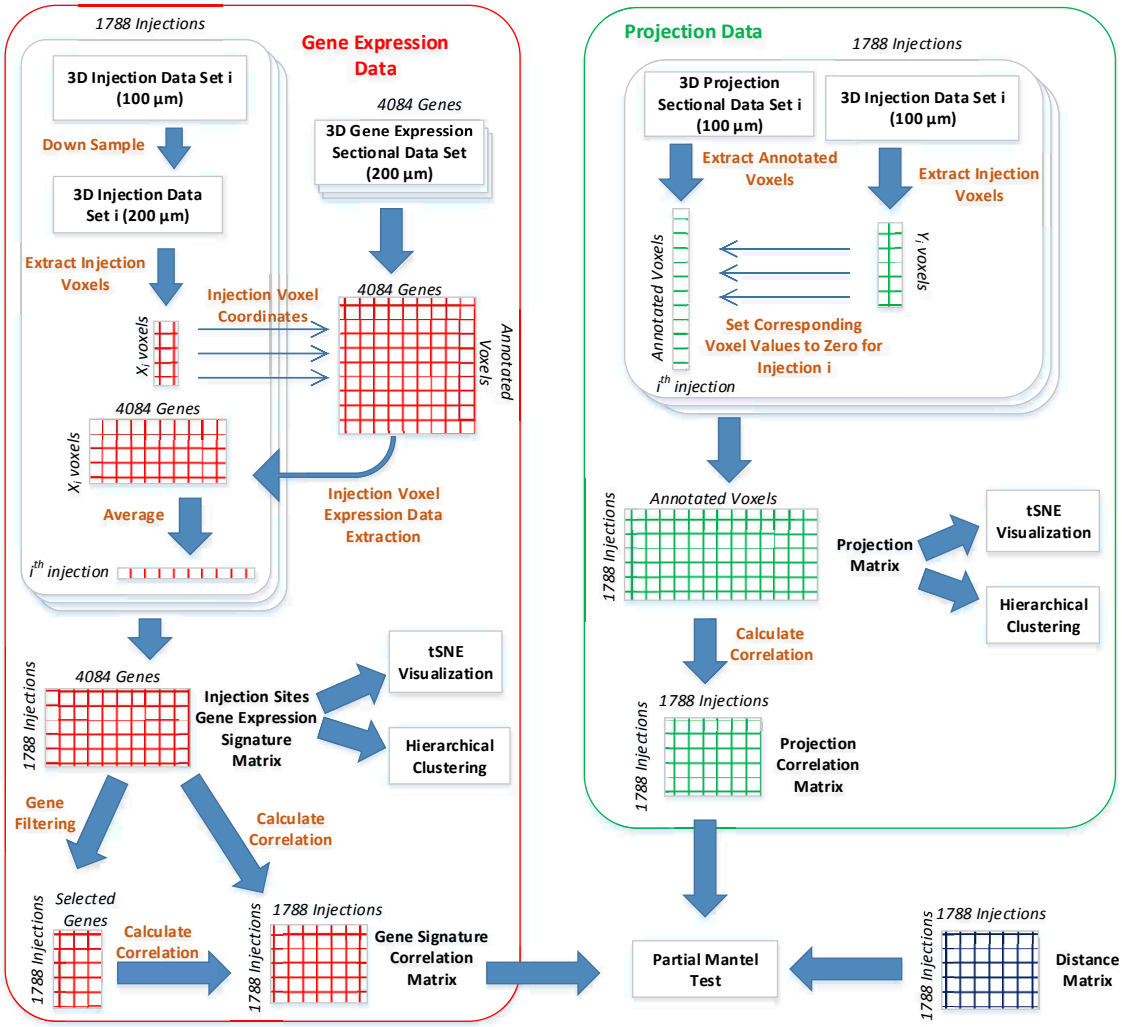


FIG. 21: Illustration of the data extraction and processing pipeline. The left and right panels show the steps involved in processing the gene expression and connectivity data, respectively. The processed data were used along with a distance matrix to perform the partial Mantel test.

derived to reflect the original data points that are far apart. It has been shown that t-SNE is able to preserve the local structure of the high-dimensional data points, and its objective function is particularly straightforward to optimize in comparison to the original SNE objective [122]. t-SNE has been used in the visual exploration of high-dimensional gene expression data [65].

In the context of my experiments, I aimed at mapping the high-dimensional gene expression and connectivity data associated with each injection site to 2D space. For each injection site, I generated gene expression and projection target signature vectors representing the gene expression and projection targets for each of the 1,788 injection sties. For the gene expression data, each vector contains 4,084 elements that correspond to the gene expression values of the 4,084 genes in the injection site. Similarly, each projection target vector contains 60,452 values representing the projection strength from the injection site to the 60,452 voxels in the entire brain. The gene expression and projection target vectors for all injection sites were collected into matrices, and t-SNE was applied to map these high-dimensional vectors onto 2D space for visual exploration.

7.4 HIERARCHICAL CLUSTERING

I employed hierarchical clustering to further explore the gene expression and projection target patterns. Hierarchical clustering constructs a dendrogram to represent the relations among all data points in a data set. Each leaf in the dendrogram represents an individual data point and each internal node represents a cluster. Such clustering method is particularly useful when the number of clusters is unknown. There are two common approaches for performing hierarchical clustering. The agglomerative approach begins by treating each individual data point as a cluster and successively

merges cluster pairs with minimal inter-cluster distance. This process repeats until a single cluster containing all the data points is obtained. In contrast, the divisive approach starts from a single cluster containing the entire data set and recursively split each cluster until each data point forms a single cluster.

Two important parameters in hierarchical clustering are the similarity measure between two data points and the criteria for computing the inter-cluster similarity. Hierarchical clustering uses linkage criteria to compute inter-cluster similarity. Three commonly used linkage criteria are single, complete and average linkages, which define similarity between two clusters as the minimum, maximum and average similarity between members in two clusters, respectively. For distance metrics, the cosine, Person correlation and Spearman correlation are commonly applied in hierarchical clustering.

Given that brain structures are hierarchically organized based on morphology and function, I hypothesized that constructing hierarchical clusters from voxels that contain expression and connectivity information is likely to recover similar brain hierarchical ontology. To test this hypothesis, I used agglomerative hierarchical clustering with complete linkage and Pearson correlation to construct dendrogram for both the gene expression and the projection target data.

Specifically, each injection site is associated with a gene expression vector and a projection target vector. These vectors are treated as individual data points. The matrix containing all the injection site gene expression vectors was used in gene expression clustering while the matrix containing all the injection site projection vectors was used in connectivity clustering.

7.5 PARTIAL MANTEL TEST

I generated the gene expression correlation matrix and the projection target correlation matrix from the injection site gene expression and projection target data matrices, respectively. I am interested in studying the relationship between gene expression patterns and projection target correlations by integrating those two correlation matrices.

Mantel test [95] determines the statistical significance of the correlation between two correlation matrices, and is a tool that matches my need. My experimental results indicate that both gene expression and project target are strongly correlated with spatial distance. That is, nearby injection sites tend to express similar sets of genes and also tend to project to similar targets. To account for spatial autocorrelation, I performed the partial Mantel test [83, 108] in which the spatial effect is excluded. Since both gene expression and projection target correlate significantly with the injection site physical distance, partial Mantel test becomes essential when studying their correlation together.

To perform partial Mantel tests, I generated a distance matrix capturing the pairwise distance between all injection sites. Specifically, I first computed the coordinate of each injection site by averaging the coordinates of all voxels belonging to that injection site. I then calculated the Euclidean distance between each pair of injections based on the averaged coordinates. I also tried using the log of the Euclidean distance, and this resulted in very similar results. The resulting distance matrix was used along with the gene correlation and projection correlation matrices to perform the partial Mantel test. This test determines the statistical significance of results by computing the p-value. Specifically, the data were randomly permuted 1,000 times and the p-value is computed as the probability that the same or higher

correlation value is achieved by the randomized data.

7.6 GREEDY GENE SELECTION

The injection site gene correlation matrix described in Section 7.5 was computed based on the correlation of all genes in the coronal set. Since not all genes contribute equally to the correlation with projection targets, I employed greedy strategies to identify subsets of genes that correlate most with the projection targets. Essentially, I aimed at removing some columns of the injection site gene signature matrix before the correlation matrix was generated. I used two greedy techniques to obtain a gene ranking that can help eliminating the least important genes.

In the greedy group gene selection approach, I followed a greedy method used in [71]. This method operates in an iterative way. In each iteration, I computed a score for each gene as the Mantel test value after eliminating its corresponding column from the injection site gene signature matrix. This score indicates the importance of each gene in determining the correlation with projection targets. After the scores for all genes were computed, a specific percentage of the least important genes were then eliminated as a group from the data set before proceeding to the next iteration. This operation continued until a predefined number of genes were obtained.

To make the greedy approach more robust, this procedure was repeated multiple times using 50% of the data randomly sampled from the original set each time. I then constructed a frequency vector for all the genes containing the frequency that each gene was selected among the multiple repetitions. Note that a similar approach was first used in [71], but the goal was not to obtain a gene ranking. I modified this technique and increased the number of repetitions and decreased the sampling percentage to obtain a gene frequency ranking. I refined my gene frequency ranking

by combining the results generated from applying this procedure several times with different parameters. I used different numbers of repetitions, different stopping criteria.

I also employed a greedy single gene selection approach as in [44] to obtain a complete gene ranking for all the genes used in my experiments. Similar to the group selection method, I computed a score for each gene in each iteration of the method to capture its effect on the correlation with the projection targets. In each iteration, only the least important gene was removed. This procedure continued until all genes were eliminated. By treating the gene that was removed first as the least important gene, I can obtain a complete gene ranking from this method. In comparison with the group selection method, the single gene selection method is much more computationally expensive. I used a parallel implementation for this scheme in order to accelerate the computation.

7.7 GENE EXPRESSION AND PROJECTION TARGETS VISUALIZATION

In this section, I report the results of visualizing the gene expression and connectivity target data by projecting them onto 2D space using t-SNE. I then performed hierarchical clustering on these two data sets to gain further insights in the next section. The primary aim of this work was to provide an integrative analysis of these two data sets and study their relationships.

I used t-SNE to visualize the gene expression and projection target data. The gene expression matrix contains 1,788 rows, and the columns represent all the genes. t-SNE was used to reduce the number of columns to 2, thereby facilitating data visualization. Similarly, the projection target matrix was also reduced to 2D. I associated each injection data set with its primary injection structure and used the same color code

provided by the ABA for visualization. The ABA color code assigns each brain structure a unique color, where nearby structures are given similar colors. The color code used in visualization is provided in Supplemental Figure 1 as in [40].

The visualization of gene expression data is given in Figure 22. I can observe that voxels with similar colors were mapped to nearby locations. This shows that gene expression patterns correlate strongly with spatial distance, a result consistent with prior findings [14, 53, 65, 67, 74]. Specifically, voxels were mainly separated into two groups, namely the brain stem and the cerebrum. In brain stem, voxels of substructures of interbrain, midbrain and hindbrain were grouped together. In cerebrum fmy major groups were observed: visual cortex, sensory-motor cortices and the rest of cortex areas, cerebral nuclei and hippocampal formation.

Unlike the gene expression data results, t-SNE visualization of the projection target data in Figure 23 was unable to show clear boundaries between brain structures. Nevertheless, this result shows that interbrain, midbrain and hindbrain structures from brainstem were still largely preserved. Although cerebrum voxels were more scattered in 2D space in comparison to those of the brain stem, some spatial structures were observed for visual cortex, sensory-motor cortices and hippocampal formation. I also observed that some voxels from brain stem were mixed with those from cerebrum. This could reflect similar connectivity patterns between them due to their characteristics in terms of neuronal information processing. For example, thalamus relays information between subcortical nuclei and the cerebral cortex. Therefore, the connectivity of voxels from cerebral cortex remains similar to those of thalamus being connected to them.

My results illustrate that the t-SNE projection of gene expression data showed a high consistency with the neuroanatomy. Similar colors representing nearby regions were mapped to nearby locations, forming clusters that are similar to the brain

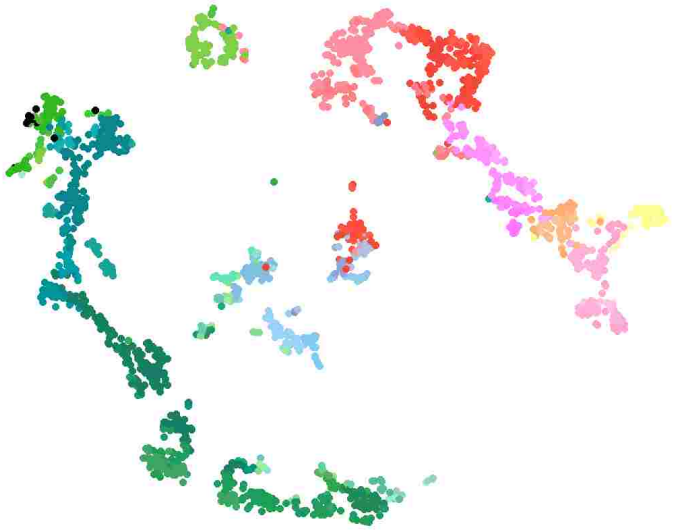


FIG. 22: Scatter plot visualization of the injection site gene expression data after mapping to 2D space using t-SNE. Each injection site is associated with its primary injection structure. The colors of structures were obtained from the Allen Brain Atlas, where similar colors represent related brain structures. The complete color code used in visualization is provided in Supplemental Figure 1.

anatomy. This indicates that the gene expression data clearly demonstrate a strong spatial locality. A similar relationship also holds for the projection target data, but to a less extent. These results indicate that both gene expression and projection target patterns exhibit spatial locality with different levels of significance.

I used the gene expression and connectivity data matrices directly in hierarchical clustering. The gene expression matrix contains 1,788 rows representing all the injection sites and 4,084 columns representing all the genes. Similarly, the connectivity data matrix contains 1,788 injection data sets as rows and 60,452 brain voxels as

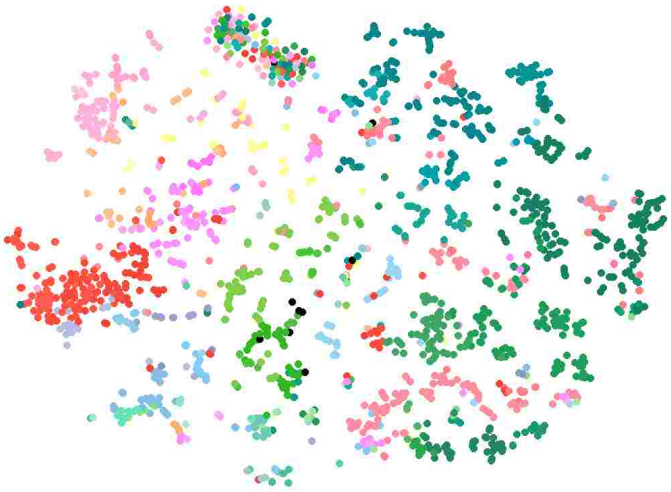


FIG. 23: Scatter plot visualization of the injection site projection target data after mapping to 2D space using t-SNE. Each injection site is associated with its primary injection structure. The colors of structures were obtained from the Allen Brain Atlas, where similar colors represent related brain structures. The complete color code used in visualization is provided in Supplemental Figure 1.

columns. I used agglomerative hierarchical clustering with complete linkage and Pearson correlation as the similar measure on both data sets. The same color code provided by the ABA was used in the dendrogram.

Figure 24 shows the dendrogram for the gene expression data set. Similar to the t-SNE visualization result, hierarchical clustering on gene expression data resulted in two major clusters, namely the brain stem and cerebrum. The voxels of interbrain, midbrain and hindbrain largely form clusters. In the cerebrum, the clusters of visual cortex, sensory-motor cortices, auditory cortex and cerebral nuclei can be clearly

observed.

Figure 25 shows the dendrogram for the projection target data set. I can observe that this clustering generated four major clusters. The first two clusters primarily involve in sensory-motor related functions, with one of which contains voxels belonging to visual and auditory area exclusively. For the other two clusters, in addition to both containing hippocampal formation and brain stem voxels, one includes cerebellum voxels and one contains cerebral nuclei voxels. In the cerebrum, I observed that despite the voxels of cerebellar cortex tend to cluster together based on their neuronal functions, they are mixed with voxels of other subcortical nuclei and the thalamus of interbrain. Such patterns revealed in my hierarchical clustering is consistent with known neuronal connectivity and function of the thalamus. That is, thalamus is heavily interconnected with subcortical nuclei and the cerebral cortex and plays an important role as information relay center. Hence, voxels of thalamus exhibit connectivity patterns similar to those of voxels from cerebellar cortex to which they are connected.

Overall, I observed that the clusters generated from the gene expression data were more consistent with the brain anatomy than the clusters generated from the connectivity data. These results are consistent with the results of visualization. Both experiments showed that spatial locality is stronger in the gene expression data than in the projection target data.

7.8 GENE EXPRESSION AND PROJECTION TARGET CORRELATIONS

The primary aim of this study was to investigate the correlation between gene expression patterns and projection target specificity. By visualizing and clustering the gene expression and projection data sets, it is clear that both of them demonstrate

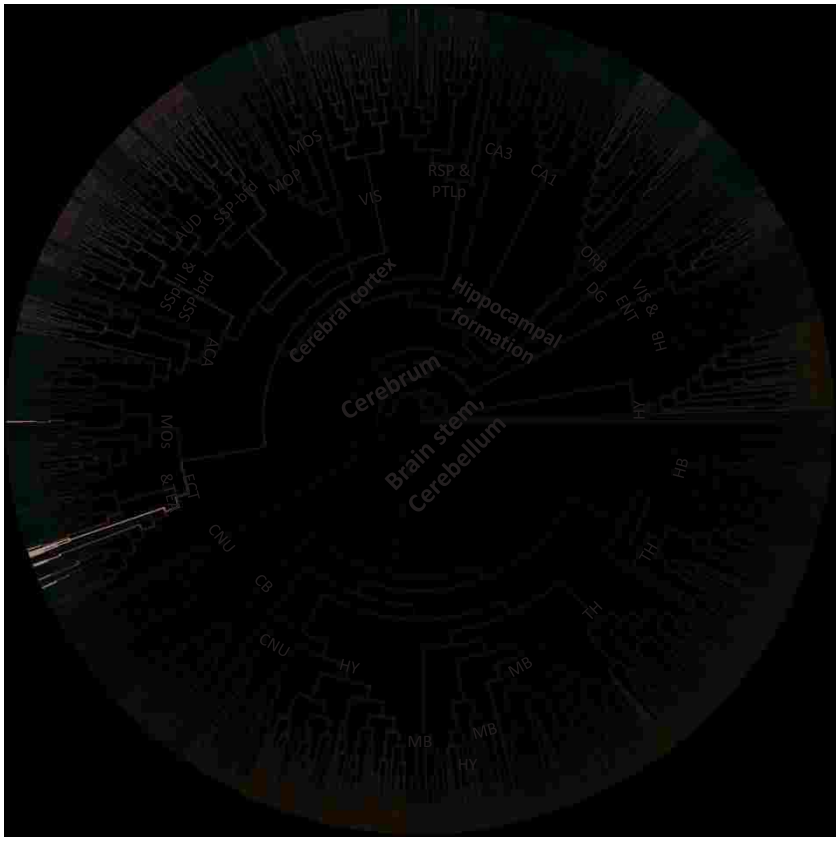


FIG. 24: Dendrogram generated by hierarchical clustering on the injection site gene expression data. The acronyms annotated on the cluster nodes were given based on the brain structure that majority of voxels in the leaf node belong to. The colors of brain structure were obtained from the Allen Brain Atlas, where similar colors represent related brain structures. The complete color code is provided in Supplemental Figure 1. The acronyms and the corresponding full brain structure names are as follows: ACA - Anterior Cingulate area, AUD - Auditory areas, CA1- Filed CA1, CA3 - Field CA3, CB - Cerebellum, CNU - Cerebral nuclei, DG - Dentate gyrus, ECT - Ectorhinal area, ENT - Entorhinal area, HB - Hindbrain, HY - Hypothalamus, MB-Midbrain, MOp - Primary motor area, MOs - Secondary motor area, ORB - Orbital area, RSP - Retrosplenial area, SSp-bfd - Primary somatosensory area, barrel field, SSP-ll - Primary somatosensory area (lower limb), TEa - Temporal association areas, TH -Thalamus, VIS - Visual areas, PTLp - Posterior parietal association areas.

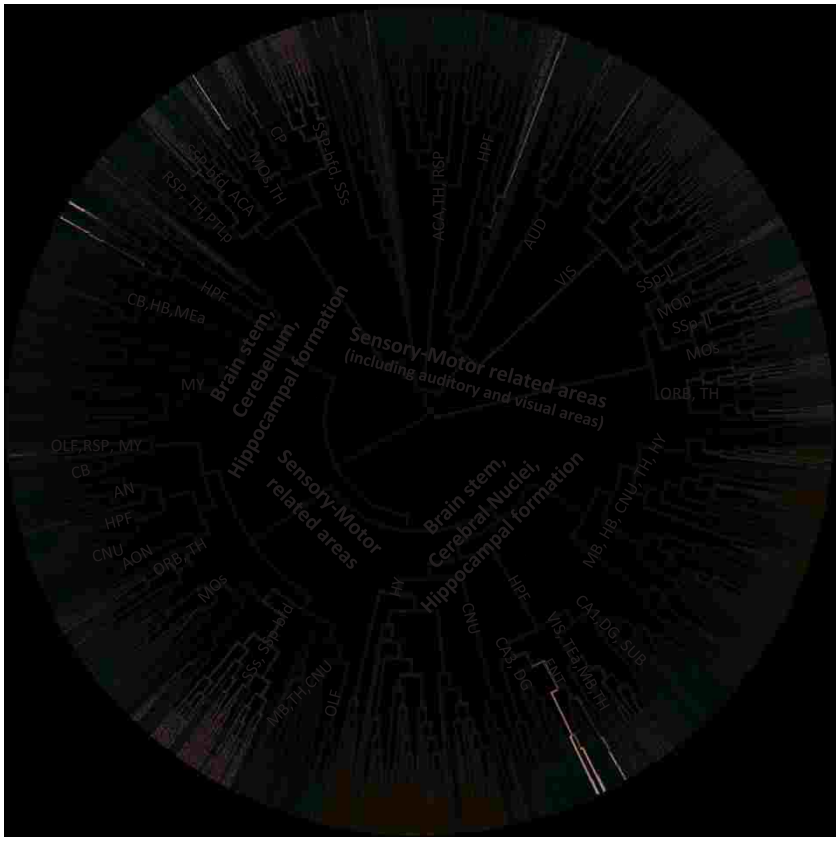


FIG. 25: Dendrogram generated by hierarchical clustering on the projection target data. The acronyms annotated on the cluster nodes were given based on the brain structure that majority of voxels in the leaf node belong to. The colors of brain structure were obtained from the Allen Brain Atlas, where similar colors represent related brain structures. The complete color code is provided in Supplemental Figure 1. The acronyms and the corresponding full brain structure names are as follows: ACA - Anterior Cingulate area, AUD - Auditory areas, AN - Amygdala nuclei, CA1- Filed CA1, CA3 - Field CA3, CB - Cerebellum, CNU - Cerebral nuclei, DG - Dentate gyrus, ENT - Entorhinal area, HB - Hindbrain, HY - Hypothalamus, MB-Midbrain, MOp - Primary motor area, MOs - Secondary motor area, OLF - Olfactory areas, ORB - Orbital area, RSP - Retrosplenial area, SSp-bfd - Primary somatosensory area, barrel field, SSP-ll - Primary somatosensory area(lower limb), TEa - Temporal association areas, TH -Thalamus, VIS - Visual areas, PTLp - Posterior parietal association areas. SUB - Subiculum, HPF - Hippocampal formation, CP - Caudoputamen.

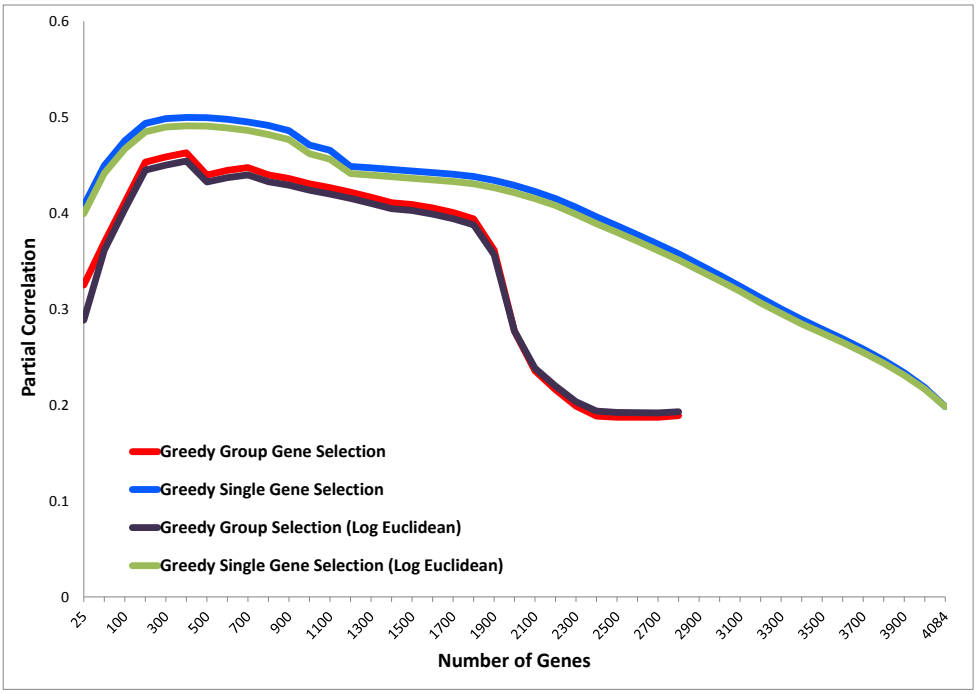


FIG. 26: The partial Mantel test results obtained by using different numbers of top ranked genes generated from two greedy gene selection approaches.

spatial autocorrelation. I therefore employed the partial Mantel test to correlate these two data sets while the spatial effect is eliminated.

I constructed the injection site gene expression correlation matrix by computing the correlation between the rows of the gene signature matrix. Similarly, the injection site projection target correlation matrix is constructed by computing the correlation between the rows of the projection signature matrix. These two correlation matrices capture the correlations between gene expression patterns and projection target

specificity in the same set of injection sites. To eliminate the spatial autocorrelation effect, I constructed a physical distance matrix that captures the pairwise Euclidean distance between the injection sites. Another distance matrix was constructed using the log of the Euclidean distance, and this resulted in very similar results. I performed partial Mantel test to quantify the significance of correlation between these two correlation matrices while eliminating their spatial autocorrelation. This test resulted in a correlation score of 0.1981 with a p-value of less than 0.001. The significance of the correlation result indicates that the gene expression patterns and projection target specificity are significantly correlated, a result consistent with the previous findings [109].

Motivated by previous studies [44, 71], I also tried to maximize the correlation score by selecting a subset of genes. Specifically, I used the greedy gene selection approaches to obtain a gene ranking and used different numbers of top ranked genes to compute the injection site gene correlation matrix. I used two greedy techniques to obtain gene rankings as described in the Material and Methods. The detailed results of the partial Mantel test corresponding to different numbers of top genes is shown in Figure 26.

It is clear from the result that the partial Mantel correlation can be significantly improved when a subset of selected genes were used. The two gene selection methods yielded a single peak approximately when the top 400 genes were used in computing the gene expression correlation matrix. The correlation scores obtained by using the top 400 genes were 0.4998 and 0.4629 for the single and group selection techniques, respectively. These scores are much higher than the score obtained by using all the genes. I note that all the p-values corresponding to the results in Figure 26 are less than 0.001 and thus are significant. The result also shows that the single gene selection technique yielded higher correlation results than the group selection method.

This is reasonable as the group selection method might exclude a batch of highly important and less important genes simultaneously as they had similar rankings at a specific iteration. On the other hand, the single gene selection technique re-evaluates all the remaining genes at every iteration after excluding one gene at a time. While both techniques had a single peak at approximately 400 genes, after closely examining those genes, I found that they only overlap in 89 genes which accounts for 22% overlap. This indicates that the high correlation obtained is not attributed to individually important genes but rather to gene groups.

CHAPTER 8

CONCLUSIONS

The key contribution of this thesis is providing several novel techniques and rich experimental analysis towards a better understanding of how the brain works. I focused my experiments on pushing the state-of-the-art techniques for automatic data processing and brain image segmentation to approach or exceed in some cases the human accuracy. I also provided multiple novel examples of experiments that could be applied to brain data to perform detailed global data modalities analysis.

In Chapter 3 of my thesis, I provided novel insights on the design of a deep neural network for brain EM image segmentation which achieved record-breaking performance. My architecture is deep, wide and carefully designed to achieve the full performance power of DNN. I achieved the best results in an ongoing online challenge without having to combine several models together, which makes the approach even more efficient. I believe that pre-training of my model on other data sets and combining the results of several models would further improve the accuracy. In addition, my model can generalize easily to different EM segmentation tasks due to the nature of DNNs, which learn features from data.

In Chapter 4, I proposed a novel technique for EM image segmentation by obtaining dense predictions that combined multi-scale contextual reasoning along with full-resolution reconstruction. My approach achieved promising performance while relying on minimum post-processing. I expect better probability maps could be generated with improvement in the z-dimension resolution of the data provided. My method can be paired with any other post-processing technique, leading to an

overall performance improvement. I did not use any hand-crafted features either in the network training or post-processing phases. Consequently, I demonstrated the ability of this model to generalize by applying it to multiple datasets obtained from different species for which it showed a consistently promising performance. I believe this method can generalize well to other similar segmentation tasks.

In Chapter 5, I investigated the relationship between gene expression and structure-level connectivity in the mouse brain. I employed two types of ensemble models, i.e., ensemble of trees and ensemble of sparse models, for predicting brain connectivity using gene expression data. My results showed that gene expression is predictive of connectivity in the mouse brain when the connectivity signals are discretized. In addition, I showed that the expression data for a small number of genes can achieve almost the full predictive power, indicating that a small number of genes are responsible for generating brain connectivity in each structure.

In this thesis, I predicted the connectivity of brain structures independently. However, the brain structures are related in a hierarchical manner in the Reference Atlas. An interesting future research direction is to incorporate such information by employing multi-task learning in which the connectivity of multiple structures is predicted simultaneously [88, 101, 134]. Furthermore, I aggregated the voxel-level data to the structure level and predicted brain connectivity at the structure level in this work.

In Chapter 6, I studied the relationship between gene expression and connectivity in the mouse brain at the voxel level. I found that the transcriptome is predictive of connectome with high accuracy. The predictive ability of the gene expression can be achieved by using only a small number of genes. The highly predictive genes are analyzed based on their Gene Ontology cellular component and biological process, and the enriched terms are indicative of their expected functions. I also showed that

the highly predictive genes are expressed more in neurons than in glia.

My work represents the first attempt to study the relationship between gene expression and connectivity in the mammalian brain at cellular resolution. In my experiments, I mainly focused on predictive analysis. I will explore other types of analysis such as the correlative and network analysis in the future.

Chapter 7 represents the first global analysis of the gene expression and projection target correlations in the adult mouse brain. I studied each modality separately and revealed their own characteristics to set the stage for the integrative study. I showed through visualization and clustering that both the gene expression and the projection targets data demonstrated significant levels of spatial autocorrelation that needs to be accounted for in the integrative analysis. By using the partial Mantel test, I showed that these two modalities were significantly correlated even after correcting for spatial autocorrelation. I employed greedy gene selection technique and used it to generate gene rankings. Based on the gene ranking results, I obtained much higher correlations by using different numbers of the top genes. The correlations results reported in this study are more significant than the values reported in previous studies given that the spatial autocorrelation effect has been eliminated.

The work presented in Chapter 7 is one of the first studies towards exploring the correlation of gene expression patterns and projection target specificity at a brain-wide scale. Given that the gene expression and the projection targets are highly correlated, a lot more in-depth analysis in this area could be further pursued. I will explore different patterns of gene expression that result in specific projection target patterns in the future. I will also perform in-depth analysis on the top genes identified in this study and investigate their functions. I will investigate whether this type of correlation between gene expression patterns and projection target specificity holds in other brains such as the human brain.

REFERENCES

[1] Allen Institute for Brain Science. Allen Brain Atlas API, 2012. URL http://www.brain-map.org/api/index.html.

[2] Allen Institute for Brain Science. Allen Mouse Brain Connectivity Atlas [Internet], 2012. URL http://connectivity.brain-map.org/.

[3] Allen Institute for Brain Science. Allen Mouse Brain Atlas [Internet], 2012. URL http://mouse.brain-map.org/.

[4] Allen Institute for Brain Science. Allen Mouse Brain Atlas: Technical White Paper: Informatics Data Processing, 2012. URL http://help.brain-map.org/download/attachments/2818169/%InformaticsDataProcessing.pdf.

[5] Allen Institute for Brain Science. Allen Mouse Brain Connectivity Atlas: Technical White Paper: Informatics Data Processing, 2013. URL http://help.brain-map.org/download/attachments/2818171/%Conn_Informatics_Data_Processing.pdf.

[6] A. Altmann, L. Tolosi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[7] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[8] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9, 2015.

[9] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[10] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[11] L. Baruch, S. Itzkovitz, M. Golan-Mashiach, E. Shapiro, and E. Segal. Using expression profiles of *Caenorhabditis elegans* neurons to identify genes that mediate synaptic connectivity. *PLoS Computational Biology*, 4(7):e1000120, 07 2008. doi: 10.1371/journal.pcbi.1000120.

[12] M. Berning, K. M. Boergens, and M. Helmstaedter. SegEM: Efficient image analysis for high-resolution connectomics. *Neuron*, 87(6):1193–1206, 2015.

[13] M. S. Boguski and A. R. Jones. Neurogenomics: at the intersection of neurobiology and genome sciences. *Nature Neuroscience*, 7(5):429–433, 2004.

[14] J. W. Bohland and *et al.* Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy. *Methods*, 50(2): 105–112, 2010.

[15] M. Bota and L. W. Swanson. BAMS neuroanatomical ontology: design and implementation. *Frontiers in Neuroinformatics*, 2(0), 2008.

[16] M. Bota, H.-W. Dong, and L. W. Swanson. From gene networks to brain networks. *Nature Neuroscience*, 6:795–799, 2003.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[18] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock. Go:: Termfinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.

[19] N. Brain Research through Advancing Innovative Neurotechnologies (BRAIN) Working Group Report to the Advisory Committee to the Director. BRAIN 2025, a SCIENTIFIC VISION. 2014.

[20] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[21] K. Briggman, W. Denk, S. Seung, M. N. Helmstaedter, and S. C. Turaga. Maximin affinity learning of image segmentation. In *NIPS*, pages 1865–1873, 2009.

[22] K. L. Briggman, M. Helmstaedter, and W. Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471(7337):183–188, 2011.

[23] P. Bühlmann. Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, pages 985–1022. Springer, 2012.

[24] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[25] J. D. Cahoy, B. Emery, A. Kaushal, L. C. Foo, J. L. Zamanian, K. S. Christopherson, Y. Xing, J. L. Lubischer, P. A. Krieg, S. A. Krupenko, W. J. Thompson,

and B. A. Barres. A transcriptome database for astrocytes, neurons, and oligodendrocytes: A new resource for understanding brain development and function. *The Journal of Neuroscience*, 28(1):264–278, 2008.

[26] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein. An integrated micro-and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS biology*, 8(10):e1000502, 2010.

[27] J. P. Carson, T. Ju, H.-C. Lu, C. Thaller, M. Xu, S. L. Pallas, M. C. Crair, J. Warren, W. Chiu, and G. Eichele. A digital atlas to characterize the mouse brain transcriptome. *PLoS Computational Biology*, 1(4):e41, 09 2005.

[28] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[29] H. Chen, X. J. Qi, J. Z. Cheng, and P. A. Heng. Deep contextual networks for neuronal structure segmentation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv preprint arXiv:1412.7062*, 2014.

[31] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, pages 2843–2851, 2012.

[32] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012.

[33] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237, 2011.

[34] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *MICCAI*, pages 411–418. Springer, 2013.

[35] I. Cribben, R. Haraldsdottir, L. Y. Atlas, T. D. Wager, and M. A. Lindquist. Dynamic connectivity regression: Determining state-related changes in brain connectivity. *NeuroImage*, 61(4):907 – 920, 2012.

[36] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013.

[37] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2012.

[38] M. de Brecht and N. Yamagishi. Combining sparseness and smoothness improves classification accuracy and interpretability. *NeuroImage*, 60(2):1550 – 1561, 2012.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[40] H. W. Dong. *The Allen reference atlas: A digital color brain atlas of the C57Bl/6J male mouse*. John Wiley & Sons Inc, 2008.

[41] H.-W. Dong, L. W. Swanson, L. Chen, M. S. Fanselow, and A. W. Toga. Genomic–anatomic evidence for distinct functional domains in hippocampal field ca1. *Proceedings of the National Academy of Sciences*, 106(28):11794–11799, 2009.

[42] A. Fakhry, H. Peng, and S. Ji. Deep models for brain em image segmentation: novel insights and improved performance. *Bioinformatics*, page btw165, 2016.

[43] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[44] L. French and P. Pavlidis. Relationships between gene expression and brain wiring in the adult rodent brain. *PLoS Computational Biology*, 7(1):e1001049, 2011. doi: 10.1371/journal.pcbi.1001049.

[45] L. French, P. P. C. Tan, and P. Pavlidis. Large-scale analysis of gene expression and connectivity in the rodent brain: insights through data integration. *Frontiers in Neuroinformatics*, 5(12), 2011.

[46] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[47] E. Geremia, O. Clatz, B. H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache. Spatial decision forests for ms lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011.

[48] D. H. Geschwind and P. Levitt. Autism spectrum disorders: developmental disconnection syndromes. *Current Opinion in Neurobiology*, 17(1):103–111, 2007.

[49] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. *arXiv preprint arXiv:1302.1700*, 2013.

[50] L. L. Glickfeld, M. L. Andermann, V. Bonin, and R. C. Reid. Cortico-cortical projections in mouse visual cortex are functionally target specific. *Nature Neuroscience*, 16(2):219–226, 2013.

[51] P. Goel, A. Kuceyeski, E. LoCastro, and A. Raj. Spatial patterns of genome-wide expression profiles reflect anatomic and fiber connectivity architecture of healthy human brain. *Human Brain Mapping*, 2014.

[52] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, D. Rueckert, A. D. N. Initiative, et al. Random forest-based similarity measures for multi-modal classification of alzheimer's disease. *NeuroImage*, 65:167–175, 2013.

[53] M. Hawrylycz, A. Bernard, C. Lau, S. M. Sunkin, M. M. Chakravarty, E. S. Lein, A. R. Jones, and L. Ng. Areal and laminar differentiation in the mouse neocortex using large scale gene expression data. *Methods*, 50(2):113–121, 2010.

[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[55] M. Helmstaedter. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature Methods*, 10(6):501–507, 2013.

[56] M. Helmstaedter and P. P. Mitra. Computational methods and challenges for large-scale circuit mapping. *Current Opinion in Neurobiology*, 22(1):162–169, 2012.

[57] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168–174, 2013.

[58] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 857–864. 2003.

[59] ISBI. Segmentation of neuronal structures in EM stacks challenge - ISBI 2012., 2012.

[60] ISBI. 3d segmentation of neurites in EM images challenge - ISBI 2013., 2013.

[61] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *NIPS*, pages 769–776, 2009.

[62] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Supervised learning of image restoration with convolutional networks. In *ICCV*, pages 1–8. IEEE, 2007.

[63] V. Jain, S. C. Turaga, K. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Learning to agglomerate superpixel hierarchies. In *NIPS*, pages 648–656, 2011.

[64] S. Ji. Computational network analysis of the anatomical and genetic organizations in the mouse brain. *Bioinformatics*, 27(23):3293–3299, 2011.

[65] S. Ji. Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering. *BMC Bioinformatics*, 14:222, 2013.

[66] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013.

[67] S. Ji, W. Zhang, and R. Li. A probabilistic latent semantic analysis model for co-clustering the mouse brain atlas. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(6):1460–1468, 2013.

[68] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[69] M. A. Just, V. L. Cherkassky, T. A. Keller, R. K. Kana, and N. J. Minshew. Functional and anatomical cortical underconnectivity in autism: Evidence from an fMRI study of an executive function task and corpus callosum morphometry. *Cerebral Cortex*, 17(4):951–961, 2007.

[70] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. R. Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.

[71] M. I. R. E. Kaufman A, Dror G. Gene expression of caenorhabditis elegans neurons carries information on their synaptic connectivity. *PLoS Computational Biology*, 2(12):e167, 2006. doi: 10.1371/journal.pcbi.0020167.

[72] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister. Large-scale automatic

reconstruction of neuronal processes from electron microscopy images. *Medical Image Analysis*, 22(1):77–88, 2015.

[73] J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi, et al. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500): 331–336, 2014.

[74] Y. Ko, S. A. Ament, J. A. Eddy, J. Caballero, J. C. Earls, L. Hood, and N. D. Price. Cell type-specific genes show striking and distinct patterns of spatial expression in the mouse brain. *Proceedings of the National Academy of Sciences*, 110(8):3095–3100, 2013.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[76] Y. Kubota, N. Shigematsu, F. Karube, A. Sekigawa, S. Kato, N. Yamaguchi, Y. Hirai, M. Morishima, and Y. Kawaguchi. Selective coexpression of multiple chemical markers defines discrete populations of neocortical GABAergic neurons. *Cerebral Cortex*, 21(8):1803–1817, 2011.

[77] S. M. Lawrie, C. Buechel, H. C. Whalley, C. D. Frith, K. J. Friston, and E. C. Johnstone. Reduced frontotemporal functional connectivity in schizophrenia associated with auditory hallucinations. *Biological Psychiatry*, 51(12):1008–1011, 2002.

[78] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[79] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[80] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, volume 2, pages II–97. IEEE, 2004.

[81] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[82] K. Lee, A. Zlateski, V. Ashwin, and H. S. Seung. Recursive training of 2D-3D convolutional networks for neuronal boundary prediction. In *NIPS*, pages 3559–3567, 2015.

[83] P. Legendre and M. J. Fortin. Spatial pattern and ecological analysis. *Vegetatio*, 80(2):107–138, 1989.

[84] E. S. Lein and *et al.* Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, 2007.

[85] J. W. Lichtman and J. R. Sanes. Ome sweet ome: what can the genome tell us about the connectome? *Current Opinion in Neurobiology*, 18(3):346–353, 2008.

[86] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region newton method for logistic regression. *The Journal of Machine Learning Research*, 9:627–650, 2008.

[87] J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–556. ACM, 2009.

[88] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 339–348. AUAI Press, 2009.

[89] J. Liu, S. Ji, and J. Ye. SLEP: Sparse Learning with Efficient Projections. *Arizona State University*, 2009. http://www.public.asu.edu/˜jye02/Software/SLEP/.

[90] M. Liu, D. Zhang, D. Shen, A. D. N. Initiative, et al. Ensemble sparse classification of alzheimer's disease. *NeuroImage*, 60(2):1106–1116, 2012.

[91] T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen. Watershed merge tree classification for electron microscopy image segmentation. In *ICPR*, pages 133–137. IEEE, 2012.

[92] T. Liu, M. Seyedhosseini, M. Ellisman, and T. Tasdizen. Watershed merge forest classification for electron microscopy image stack segmentation. In *ICCV*, volume 2013, page 4069. NIH Public Access, 2013.

[93] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[94] A. Mahfouz, M. van de Giessen, L. van der Maaten, S. Huisman, M. Reinders, M. J. Hawrylycz, and B. P. Lelieveldt. Visualizing the spatial gene expression organization in the brain through non-linear similarity embeddings. *Methods*, 2014.

[95] N. Mantel. The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27(2):209–220, 1967.

[96] N. Meinshausen and P. Bühlmann. Stability selection. Technical report.

[97] L. Ng, S. Pathak, C. Kuan, C. Lau, H. Dong, A. Sodt, C. Dang, B. Avants, P. Yushkevich, J. Gee, D. Haynor, E. Lein, A. Jones, and M. Hawrylycz. Neuroinformatics for genome-wide 3-D gene expression mapping in the mouse brain. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:382–393, 2007. ISSN 1545-5963. doi: http://doi.ieeecomputersociety.org/10.1109/tcbb.2007.1035.

[98] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[99] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PloS ONE*, 8 (8):e71715, 2013.

[100] S. W. Oh and *et al.* A mesoscale connectome of the mouse brain. *Nature*, 508 (7495):207–214, 2014.

[101] T. K. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20 (6):3465–3489, 2010.

[102] N. Rajagopal, W. Xie, Y. Li, U. Wagner, W. Wang, J. Stamatoyannopoulos, J. Ernst, M. Kellis, and B. Ren. RFECS: A random-forest based algorithm for enhancer identification from chromatin state. *PLOS Computational Biology*, 9 (3):e1002968, 2013.

[103] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 234–241. Springer, 2015.

[104] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon. Sparse logistic regression for whole-brain classification of fMRI data. *NeuroImage*, 51(2):752 – 764, 2010.

[105] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[106] D. Sieburth, Q. Ch'ng, M. Dybbs, M. Tavazoie, S. Kennedy, D. Wang, D. Dupuy, J. Rual, D. Hill, M. Vidal, G. Ruvkun, and J. Kaplan. Systematic analysis of genes required for synapse structure and function. *Nature*, 436(7050):510–7, 2005.

[107] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[108] P. E. Smouse, J. C. Long, and R. R. Sokal. Multiple regression and correlation extensions of the mantel test of matrix correspondence. *Systematic Zoology*, 35 (4):627–632, 1986.

[109] S. A. Sorensen, A. Bernard, V. Menon, J. J. Royall, K. J. Glattfelder, T. Desta, K. Hirokawa, M. Mortrud, J. A. Miller, H. Zeng, et al. Correlated gene expression and target specificity demonstrate excitatory projection neuron diversity. *Cerebral cortex*, page bht243, 2013.

[110] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[111] A. Statnikov, L. Wang, and C. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9(1):319, 2008.

[112] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. In *NIPS*, pages 2980–2988, 2015.

[113] S. M. Sunkin, L. Ng, C. Lau, T. Dolbeare, T. L. Gilbert, C. L. Thompson, M. Hawrylycz, and C. Dang. Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucleic Acids Research*, 41 (D1):D996–D1008, 2013.

[114] L. W. Swanson. *Brain Maps: Structure of the Rat Brain*. Academic Press, San Diego, CA, USA, 3rd edition, 2003.

[115] L. W. Swanson. *Brain Architecture: Understanding the Basic Plan*. Oxford University Press, New York, NY, USA, 2nd edition, 2011.

[116] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[117] C. L. Thompson, S. D. Pathak, A. Jeromin, L. L. Ng, C. R. MacPherson, M. T. Mortrud, A. Cusick, Z. L. Riley, S. M. Sunkin, A. Bernard, R. B. Puchalski, F. H. Gage, A. R. Jones, V. B. Bajic, M. J. Hawrylycz, and E. S. Lein. Genomic anatomy of the hippocampus. *Neuron*, 60(6):1010–1021, 2008.

[118] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[119] M. Toledo-Rodriguez, B. Blumenfeld, C. Wu, J. Luo, B. Attali, P. Goodman, and H. Markram. Correlation maps allow neuronal electrical properties to be predicted from single-cell gene expression profiles in rat neocortex. *Cerebral Cortex*, 14(12):1310–1327, 2004.

[120] L. Tolosi and T. Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011.

[121] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, 2010.

[122] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[123] V. Varadan, D. M. Miller, and D. Anastassiou. Computational inference of the molecular logic for synaptic connectivity in *C. elegans*. *Bioinformatics*, 22(14): e497–e506, 2006.

[124] C. Watson, M. Kirkcaldie, and G. Paxinos. *The Brain: An Introduction to Functional Neuroanatomy*. Academic Press, San Diego, CA, USA, 2010.

[125] C. Watson, G. Paxinos, and L. Puelles. *The Mouse Nervous System*. Academic Press, San Diego, CA, USA, 2011.

[126] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 314(1165): 1–340, 1986.

[127] L. Wolf, C. Goldberg, N. Manor, R. Sharan, and E. Ruppin. Gene expression in the rodent brain is associated with its regional connectivity. *PLoS Computational Biology*, 7(5):e1002040, 05 2011.

[128] J. Ye, M. Farnum, E. Yang, R. Verbeeck, V. Lobanov, N. Raghavan, G. Novak, A. DiBernardo, and V. A. Narayan. Sparse learning and stability selection for predicting mci to ad conversion using baseline adni data. *BMC neurology*, 12 (1):1, 2012.

[129] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[130] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *The Journal of Machine Learning Research*, 11:3183–3234, 2010.

[131] G.-X. Yuan, C.-H. Ho, and C.-J. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012.

[132] L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, J. Ye, A. D. N. Initiative, et al. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61(3):622–632, 2012.

[133] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[134] D. Zhang, D. Shen, A. D. N. Initiative, et al. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *NeuroImage*, 59(2):895–907, 2012.

[135] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, 108:214–224, 2015.

[136] X. Zheng and J. C. Rajapakse. Learning functional structure from fMR images. *NeuroImage*, 31(4):1601–1613, 2006. ISSN 1053-8119.

[137] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012.

[138] A. Zlateski and H. S. Seung. Image segmentation by size-dependent single linkage clustering of a watershed basin graph. *arXiv preprint arXiv:1505.00249*, 2015.

VITA

Ahmed Fakhry
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

I received my Bachelor degree in Computer Science from Alexandria University, Egypt. In Fall 2012, I joined the Computer Science Department of Old Dominion University and started my research in machine learning, deep learning and computational biology. I have worked on analyzing the adult mouse brain project using machine learning and statistical models. My work has revealed a number of novel correlations between different modalities in the mouse brain such as the gene expression and regional connectivity. I also worked on 2D and 3D segmentation of EM brain images for the *Drosophila* and mouse where my work has achieved the top performances in several open online challenges. This work is considered the state-of-the-art segmentation techniques for Electron Microscopy images. I have already published and submitted five papers in highly-regarded journals and conferences and I serve as a primary or co-author for a total of seven papers.